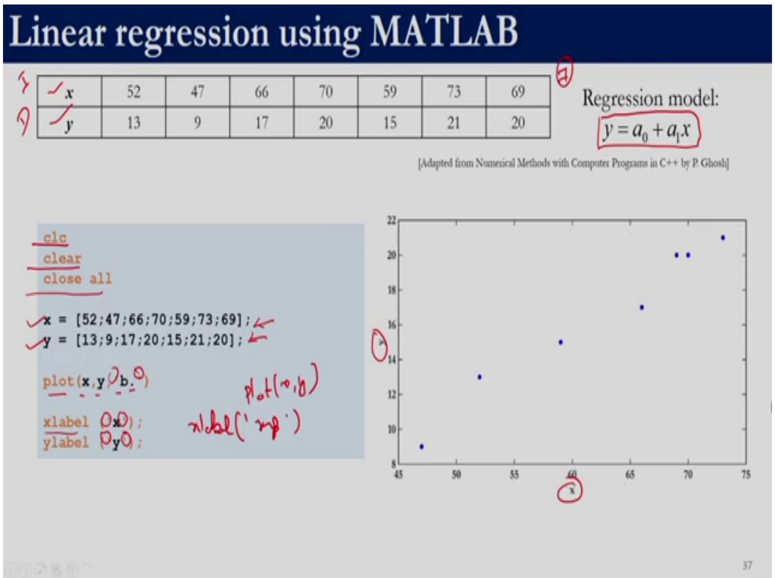
Computer Aided Applied Single Objective Optimization
Dr. Prakash Kotecha
Department of Chemical Engineering
Indian Institute of Technology, Guwahati

Lecture - 05
Curve fitting: Regression

(Refer Slide Time: 00:31)



So, far we have seen the, theory of linear regression right. So, now, we look into the implementation of linear regression using MATLAB. Again, as we have been stressing from the beginning of the course here, we are using MATLAB you are free to use any other software. So, this is the so, we will initially begin with simple linear regression problem right. So, where in we have this x and y data is given right. So, this is our independent variable, this is our dependent variable right.

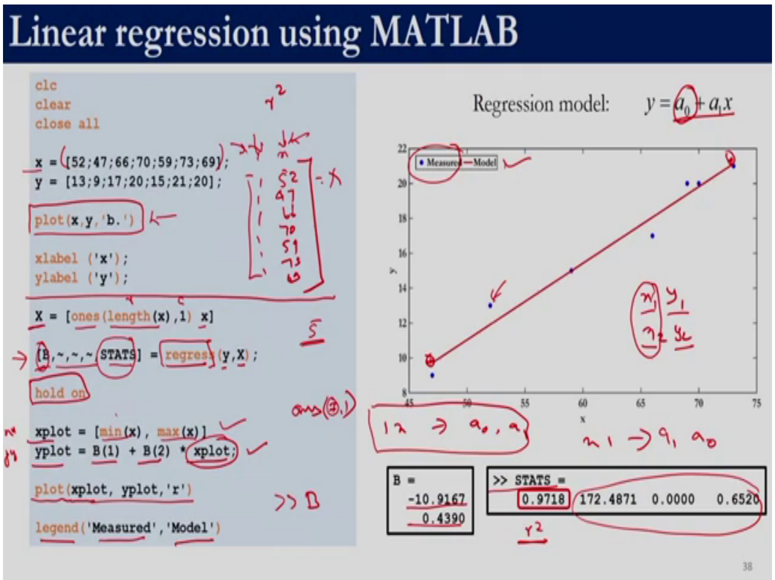
So, we have here seven points and we want to fit this model y is equal to a naught plus a 1 x right. So, in MATLAB the function that we will be using is a regress right. So, before using regress, let us just try to get this plot in MATLAB. So, as you might already know clc helps in clearing the command window, clear will help us to clear the workspace of MATLAB and close all will lead to closing all the currently open graphic windows right. So, now, we define the vector x and y right.

So, this is nothing, but just the definition. So, this semicolon, make sure that we get a column vector for x and column vector for y right and then we use the plot function. So, when we say plot x comma y, if you just say plot x comma y what would happen is, we would not get such a plot, but we would a get a plot where in all these lines are connected.

So, we do not want that to happen, we just want the points right. So, that is what we are doing is, we are saying plot x comma y comma, this again within single quotes over here, it is within single quotes. So, this b stands for blue and this dot stands for placing a dot. So, what we are asking MATLAB is to just put a blue color dot wherever the points are there right.

So, we would gets a plot similar to this and this helps us to add the label. So, this is x label. So, we just want to write x and y, for the x axis and the y axis right. So, again we give this in single quotes. So, whatever we give in single quotes will be replicated as such. So, if have given let us say x label, within single quotes if you had given temperature right. So, temperature would have been returned over here right.

(Refer Slide Time: 03:01)



So, this piece of code will help us to just get this plot right, so far we have done right. So, for linear regression the function in the inbuilt functioning of MATLAB is regress. So, for regress we need to give the dependent variable comma independent variable right. So, independent variable, it can be multi variable also.

So, regress can do simple linear regression, multi linear regression and we can even use it to do polynomial regression right. So, we need to give independent dependent variable comma, all the independent variables. In this case we have only one independent variable x right, but the model that we want to fit is a naught plus a 1 a naught plus a 1 x right, but this regress function is based on the general linear least squares which we have discussed previously right. So, there if you remember we have to stack the independent variables with the column of ones right. So, if you our model has an intercept right.

So, in this case we want to fit a model which has an intercept. So, what we need to do is; we need to give a column of ones. So, this is what we want to construct right since there are 1, 2, 3, 4, 5, 6, 7 data points, we want this column of ones right with 7, 7 rows right. So, to get that we will use this ones function of MATLAB, o n e s; so, ones function of MATLAB can be used to generate as many rows and columns that are required.

So, right now we can actually say 7 comma 1, but if we change this x with 10 values, then again we need to go and do this right. So, instead of writing this number 7 what we will say is ones of length of x. So, even if this vector x changes, it will measure the length and then it will appropriately select right. So, that is what we are doing ones of the number of rows that we want is nothing, but length of x, the number of columns that we want is one one column right, because this is what we want.

So, that so, this is number of rows, this is number of columns and this is the function. So, this will just give us a column of ones, how many rows seven rows, because the length of x is 7 and then we are stacking this x vector. So, basically what we are doing is 152, 147, 166, 170, 159, 173 and 169 ok. So, this is what we are constructing as capital X as upper case x right. So, once we have constructed this, we are ready to use the regress function.

So, here we need to give regress y comma x. So, this regress function can return a large number of values, all of them have their own significance, but in this case, but in our course we have restricted our self to finding out this model coefficient and the coefficient of determination r square right. So, basically regress actually gives us 1 2 3 4 5 output arguments. The first argument is the coefficients right. So, it is not necessary to write B if we can write any variable name right.

So, we have chosen to write B you can write any variable and these three arguments, the next three arguments, because of whatever we have in the whatever we have covered in this course we are not in a position to interpret that. So, we are not receiving those values, but if you wanted those values you can just say B comma, C comma, D comma, E or whatever appropriate variable names and stats. So, stats is again a variable name you could have given any other variable name over there right. So, when we do this.

So, this line will help us to solve the regression problem and the coefficients would be returned and stats will be a vector right and the first value in the vector stats will correspond to the r square value right. So, stat stats is going to give us a number of values. The first one is what we are interested what we are interested in right. So, once we have obtained the coefficients and the regress function right, we also want to plot the line.

So, as we had seen previously in that Anscombe's data for at least for linear regression, simple linear regression, it is always a good idea to have a look at the plot right rather than merely relying on r square value right. So, in order to plot that what we want is we want the points as well as the straight line right. So, we want something similar to this right. The points are also, the points also need to be there and we also need the line.

So, what we are going to do is. So, since we are going to plot on the same plot we are going to use this hold on right so; that means, it will plot on the same plot where we had plotted the data points right. So, now, how do I get the straight line. So, for straight line we need two points right, if we have two points we can draw a straight line so, x 1 y 1 and x 2 y 2. So, what we are going to do is so, this y 1 and y 2 is from the model right once, this regression problem is solved the model coefficients are in this variable B.

So, what we will do is for x 1 and x 2 we will say the minimum value in this vector is going to be our x 1. So, that is what I am we are doing over here. So, the minimum value of x is going to be our x 1 and the maximum value of x is going to be our x 2. So, the line is going to span from this point till this point right. So, once we have decided on the x values, x 1 x 2, we can calculate y 1 and y 2, because we know the model coefficients B right.

So, this model coefficient will have as many elements as many columns in this x matrix right. So, in our case we have two columns. So, it will consists of two values right. The first value would be the constant value, because we have stacked one over here and the independent variable x over here. So, it is 1 comma x. So, this coefficient of x is will be the second element and the constant coefficient will be the first element right, had we interchange these two columns the values in b what we have got would also have would also be in that order.

So, we would not get a naught comma a 1 we would get a 1 comma a naught right. So, if we give one comma x we are going to get a naught comma a 1 right, but if we had given x and 1 the coefficients would have been in the order a 1 and a naught right. So, right now this is the convention that we have used right. So, the model is actually b of 1. So, the first value of 1 plus the second value of B and multiplied by the x values right. So, x x is a vector now right.

So, we will get two values for y plot right again is this x plot and y plot are just variable names you could have even given xx yy right. So, since we are going to use it for plotting we just wrote x plot and y plot right. So, now, we have calculated the x 1 x 2 over here, it will directly calculate y 1 y 2 it will calculate two values, because x plot is a vector right.

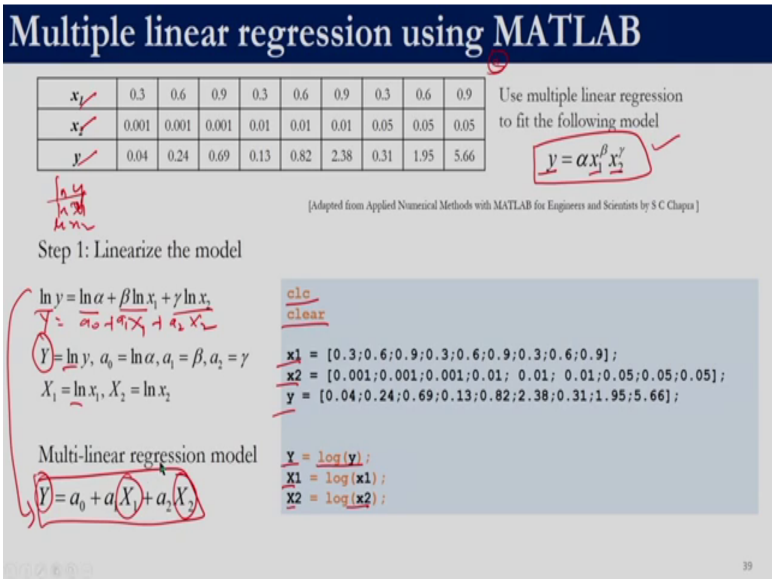
So, then we can plot we can use this plot function directly to say plot x plot comma y plot right with a red color line right. So, here we are not saying that r dot, if we had said r dot it would have plotted this particular point corresponding to this and this particular point corresponding to the end right, but since we wanted the entire line right. We are giving just r right, we are not specifying any symbol next to r right. So, it will plot a red color line.

So, once we are done with that we will be able to see the points as well as the model on the same plot right and then we can add a legend right. So, legend so, remember MATLAB is not going to decide whether this is the data point or this is the model right. So, we need to specify that. So, since we started with by plotting the points right, we say the first thing that we plotted is measured. So, it is going to take this blue dot for measure and then the second thing which we plotted was the model right. So, we write measured comma model within single quotes separated by a comma.

So, we will get this legend also right. So, that is how we plot and if we type just b in the command window right we will get these two values minus 10.9167 and 0.4390 similarly, if we type stats in the command window or any variable name that you had used over here we will get this four values right, all these values do have their own significance, but since we have not covered that as part of this course, we will only explain, what is the first value. So, the first value stands for the coefficient of determination r square right.

So, as you can see it is very simple to use the regress function for simple linear regression. Again, as we will show even for multi linear regression the procedure is rather very straight forward right. So, now, let us look into how to solve a, multiple regression problem using the same regress function in MATLAB right.

(Refer Slide Time: 12:09)



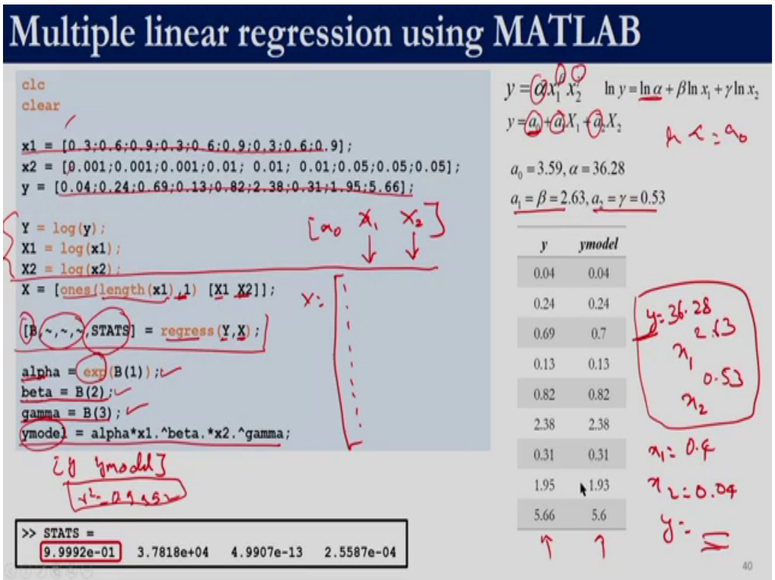
So, here we have this problem where in the model that we have to fit is this, y is equal to alpha x 1 power BETA x 2 power gamma right. So, this is a non-linear model right either we can directly solve this as a non-linear regression problem or we can use any one of the data transformation technique that we have seen previously. So, y is the dependent variable, x 1 and x 2 are the independent variable.

So, x 1 x 2 are the independent variable and y is the independent variable right. So, here we have 1 2 3 4 5 6 7 8 9, 9 points right. So, what we will first do is, we linearize this model right. So, to linearize this, we can take lon on both sides right. So, this equation becomes ln y is equal to ln alpha plus BETA ln x 1 plus gamma ln x 2 right. So, this is the linearized form right. So, since we know y ln y can be calculated right. So, ln y we will represent it by this notation y right. So, and this one if we see this is a naught right plus a 1 ln x 1 can be calculated. So, we will call this as x 1 plus gamma let us say that is a 2 is capital X 2, because we know this x 2.

So, we can calculate ln x 1 and ln x 2. So, once we have calculated that this model becomes this one right. So, now, this if you see it is a typical linear regression model, which we have discussed previously. So, first for implementing in MATLAB we will have to specify the data points rights. So, clc is again to just clear the command window clear will help us to clear the workspace. So, we are defining these three points right. So, ln y ln x 1 ln x 2 we are not calculating we will just use MATLAB to calculate that right.

So, we have defined all the three vectors right, then we say that the uppercase y, the variable y is nothing, but log of y. So, remember in MATLAB the natural logarithm is actually log right whatever I was referring whatever we have referred as ln is actually log in MATLAB right. So, the variable capital X 1 is log of the variable x 1 and capital X 2 is log of x 2 right. So, now, we have all the data's in the transformed space. So, we have this y; we have this x 1; we have this x 2 ok

(Refer Slide Time: 14:45)



So, then we will do the same thing right. So, till this we have discussed right. So, here our model actually has a coefficient right and the regress function needs the independent, the dependent variable comma all the dependent variable does not matter whether, we have 1 independent variable, 2 independent variable or 100 independent variable. All of them have to be stacked right the rows are the data points, the columns are the independent variables right.

So, in this case what we actually want as axis. So, we will define the convention right. So, for this code the convention is the constant term a naught and then we will have a one the x 1 coefficients and the x 2 right. So, for a naught we will have to stack a column of ones, we will have to have a column of ones. So, how many ones do we need, depends on the length of x 1 right. So, the same thing what we used previously; so, this will calculate the length of x 1, it will generate that many rows, because and one column right.

So, this will just give us a column of ones and then we stack the variable x 1 and the variable x 2. Remember, the variable upper case x 1 and uppercase x 2 right. So, now, we are working with this three and then we give it into regress right. Once, we execute this code again, we will get those five variables; these three variables again we do not we have not seen how to interpret this. So, we will get the coefficients in the variable B and the statistical parameters in the variable stats right and we are interested only in the first value of stats, because that corresponds to r square right.

So, once we have determined that right here we are not going to plot it, because we have x 1 x 2 and y right. So, we will not plot it right, but our task was to actually find out what is alpha BETA and gamma what we have found out is a naught a 1 and a 2 right. So, if you remember the transformation which we did that a one actually corresponds to BETA and a 2 actually corresponds to gamma right.

So, BETA is nothing, but the second value gamma is nothing, but the third value second value and third value, because x 1 is the second column and x 2 is the third column right. So, alpha is actually we have used alpha is equal to so, in previous slide if you have seen this is ln alpha is actually a naught. So, ln alpha is a naught right. So, now we have a naught we want l alpha. So, we take the exponential of that right. So, that can be done using the exp function of MATLAB right. So, now, we have alpha, BETA and gamma right. So, we have got the parameters which we which we had set out to find right, once that is done since, I am not we will not be able to plot.

Let us actually see that for each of this x 1 and x 2 what would my model give right. So, that can be determined by this expression right. So, this is nothing, but the values obtained by model, any variable name, we have choose as y model is equal to alpha, because that is what is our model alpha into x 1 x 1 to the power BETA multiplied by x 2 to the power BETA right.

So, this will give us the value of the model at each of this points right, whatever the number of points are there for each of those points we would get the y values from the model and we already know what are the y values which was given to us right. So, this is just a comparison of that thing right.

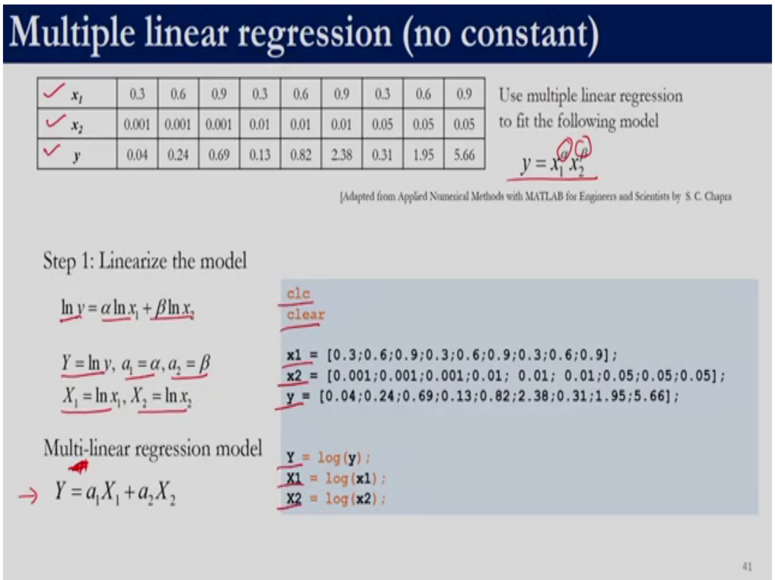
So, after this you could have just written y y model in MATLAB. So, it would have displayed something similar to this right. So, for the so, this will help us to kind of see how the fitters right since, we are not plotting over here, from here we can see right. So, more or less the fit seems to be good, because this column and this column are more or less kind of similar right. So, and we can also have a look at the first value of stats right.

So, if you just type stats over here, it will give us these four values right. So, the first value is our regression. So, this e power minus 0 1 stands for 10 power minus 1. So, the r square is actually 0.9999 2 right. So, it seems to be a very good fit right and the model is; obviously, y is equal to alpha, alpha is 36.28 right, x 1 power BETA so that is 2.63 x 2 power gamma which is 0.53.

So, this is what we will get from the model; so, for any x 1 and x 2 for which we do not have the value of y. So, for example, x 1 is equal to again, it has to be in the same domain. So, let us say x 1 is equal to 0.4 and x 2 is equal to 0.04, we can use this expression to find out the y values. So, that was one example for multiple linear regression.

Now, we look into another multiple regression problem right, but this time we will not have a constant coefficient right. So, the both the examples which we have seen so far had a constant in the model right. So, that is why we were stacking a column of ones and then we were stacking the independent variables right. So, the next model which we are going to consider is not going to have a constant coefficient right.

(Refer Slide Time: 20:29)

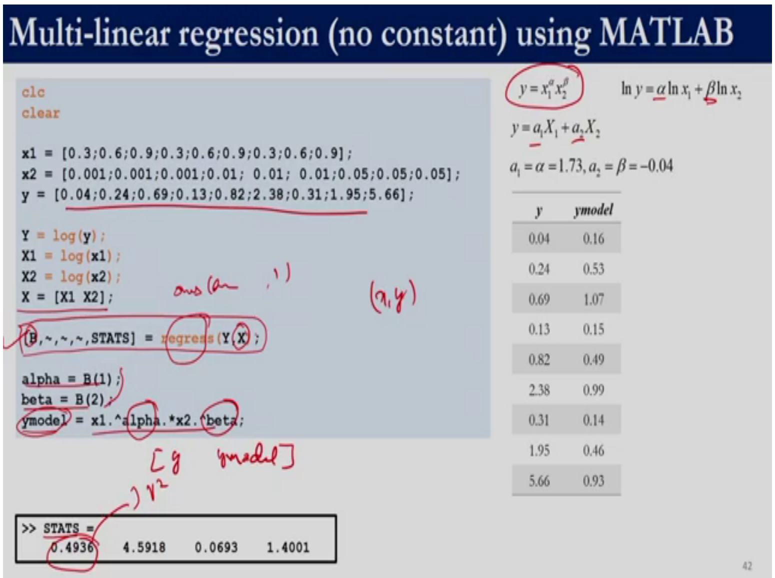


So, here again two independent variables x 1 x 2, these are the y values. So, these values are given to us and we are expected to fit this model. So, our task is to find out the values of alpha and BETA for which this model would best represent this data right ok. So, again we linearize it right. So, we will take lon, lon y is equal to alpha ln x 1 plus BETA ln x 2.

So, if we use the variable capital Y to indicate l and y a 1 for alpha a 2 for BETA capital X 1 for ln x 1 and capital X 2 for ln x 2 right. So, we basically end up with this model right. The only difference is previously we had a naught it was a naught plus right now, we do not have this constant coefficient right.

So, it is going to be exactly the same except that for determining the x matrix which has to be fed into the regress function, we will not stack it with a column of ones right. So, again these are exactly same what we have discussed clearing the workspace and the command window, defining the data, transforming the data and then we determine capital X 1 which is x 1 and x 2.

(Refer Slide Time: 21:42)



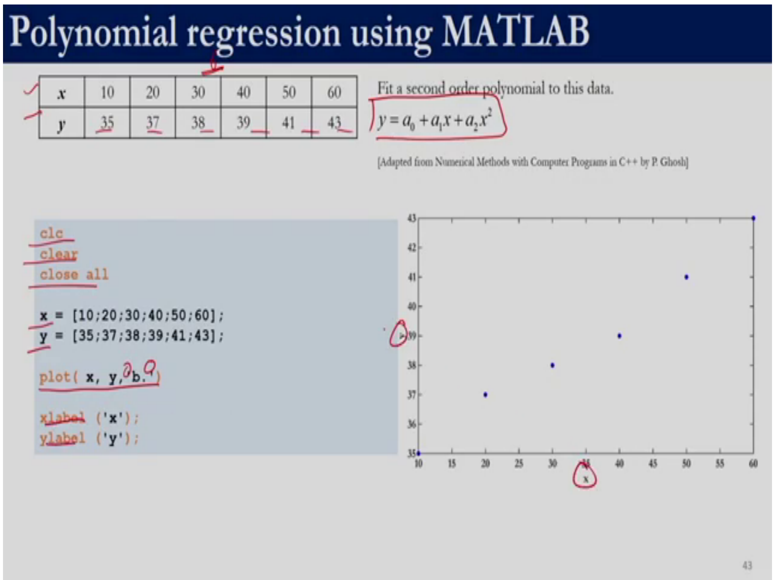
Here, we are not stacking ones of length of x 1 comma 1 right, because we do not we do not have constant term in our model itself right. So, the same way that we have been calling this regress function it is same thing whether there is a constant coefficient or not the regress function remains the same, whether there is a constant coefficient or not is to be captured you for this variable x right.

So, we give remember it is a dependent variable comma independent variable usually, we are accustomed to write x comma y if you do that you would get an error right and it would probably tell you why the reason, why you get that error. unlike the previous case wherever there were three values. Now, we will have only two values this a 1 and a 2 right and a 1 and a 2 directly correspond to alpha and BETA right there was no transformation of them.

So, this is alpha, this is BETA. So, similarly like last time we can also calculate y model again this is some variable name which we have chosen right; so, x 1 x 1 to the power alpha x 2 to the power BETA right. So, this can be calculated, because once we are done with this step right, we actually know the coefficients right and alpha and BETA are defined here right. So, we will be able to calculate the y model values right.

So, again if you do y comma, y model you will be able to see the values which are given to us y and what our model is predicting right, over here you can see that the difference is significant right in this. So, the r square it is not going to be a very high value right. So, again if you type stats in the command window the first variable, the first value corresponds to r square. So, r square in this case is 0.4936, which is not a very good value right ok.

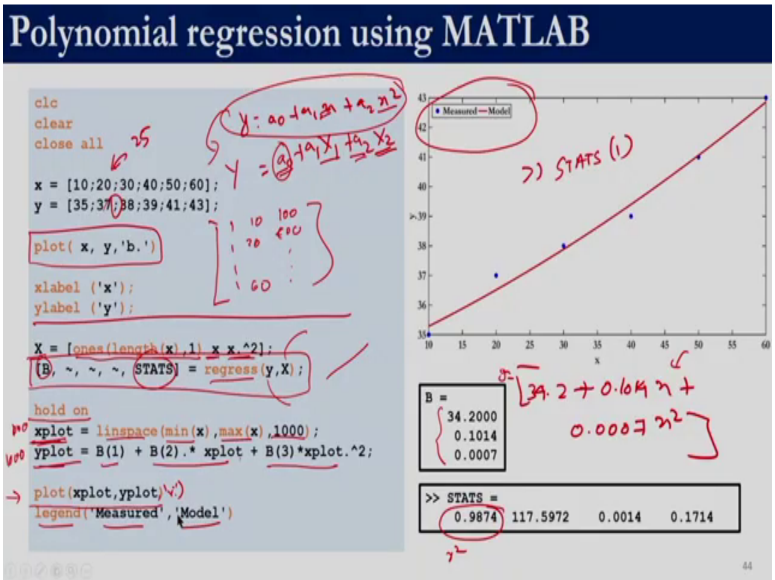
(Refer Slide Time: 23:42)



So, now that is about multi linear regression all right. So, now, let us go to polynomial regression. So, polynomial regression can be done in two ways; we can either use the regress function itself or there is a separate function called as poly fit right that can also be used right. So, we will see both of them we will start with the regress function and then we will also look into the poly fit function right. So, this is the polynomial that we are required to fit y is equal to a naught plus a 1 x plus a 2 x square and these are our data points x and y right.

So, here we have six data points right and this is the y, these are the y values ok. So, the first task is to define whatever is required. So, again clearing the command window, clearing the workspace, closing all the actively open plots, the x vector, the y vector we are just defining them and then again since we want only the points we do not want them to be connected by a piecewise linear function right. So, we just give plot x comma, y within single quotes right b dot. So, it will put a dot blue color dot at wherever the points are there and then we add the x label so, to get this x and y.

(Refer Slide Time: 24:59)



So, now, we have the data points on a plot right. So, this is what we have discussed so far. So, our model is y is equal to a naught plus a 1 x plus a 2 x square right. So, what we will do is we will say, we know how to do multi regression. So, we will convert this into multi regression we will say this is y is equal to a naught plus a 1 x 1 plus a 2 x 2 right. So, the variable x 2 is nothing, but the variable x squared right, each value squared that would be x 2 right. The capital X 1 is nothing, but the x that we have received right and since we have a constant coefficient we need to stack a column of ones right.

So, whatever this polynomial regression is there that we have converted into multi linear regression right. This we had seen even previously right. So, this part now by now you should be comfortable with we measure the length of x, create as many rows involving ones, then we have this x and we have this x square right. So, we will have three columns. So, the first column would be 1, the second next column would be 10 20 30 40 50 60 and then would be 10 square is 100, 400 and so on right.

So, that would be what would be in x right and again, this is the same way we are accessing the regress function right. These two are variable names which we have used, regress is the function dependent variable comma independent variables right. So, here we will give this x which we have constructed in the previous line and then since over here we have only x and y, we can actually plot and visualize the model right. So, since we want to do it on the same plot we do hold on right.

So, whichever plot is this active plot on the same thing we are plotting right. So, when we were doing linear regression linear regression, we needed only two points to fit a straight line right, but now we are fitting a polynomial right. So, two points are not sufficient. So, what we are doing is from the minimum value of x to the maximum value of x we are generating 1000 points linearly spaced right.

So, what we are doing this between this 10 and 60 that is the min and max between this 10 and 60 we are generating 1000 points. So, at each of this point we will use the model to find out the y value. So, that will help us to visualize the model right. So, otherwise if we are going to do two points then we will just get a straight line right, but the model is not a straight line right.

So, that is why we are generating 1000 points. So, again the first value since we know the order in which we gave these columns we also know in what order will we get the coefficients. So, the first coefficient b 1 is going to be the constant coefficient, the second one is going to be the model the coefficient of x and the third value is going to be the coefficient of x square right.

So, that is why we do b of 1. So, the first value which is constant plus b of 2 which is nothing, but a 1 multiplied by x plot x plot is the points that we generated plus b of 3. So, a 2 into x plot so, these points each of the point squared right. So, this will give us y plot. So, x plot will have 1000 points and y plot will have 1000 points.

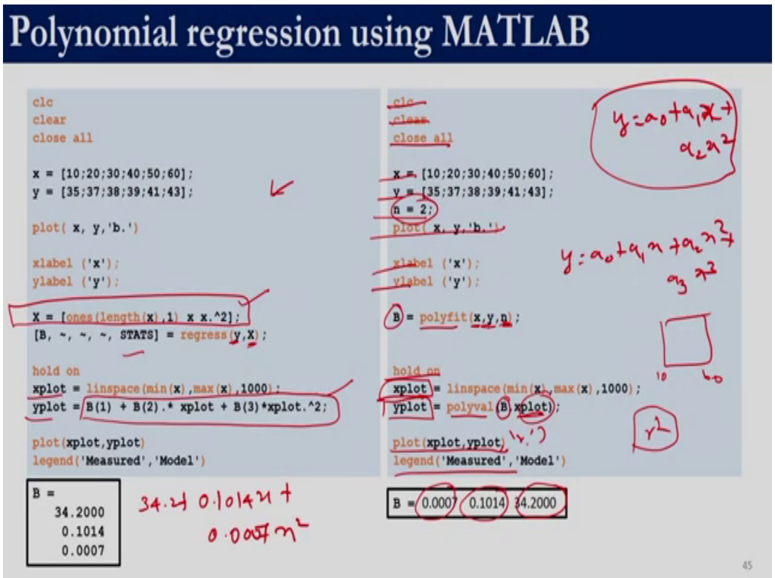
So, now, if we plot x plot and y plot these are actually 1000 points over here, but since they are very closely spaced, it seems like continuous curve right hm. So, a better way to do this was plot x plot x plot comma y plot comma let us say r dot dot and then close the bracket right within this single quotes r dot. So, we would have actually got points right.

So, that is a better way to do it than this one right. So, similarly like last time we did legend right, first we had plotted the measured values. So, we give measure and then we plotted the model. So, we give measure comma model. So, we will get this legend also over here, then we can if we type b in the command window we will get these three. So, the model is actually 34.2 right plus a 1 is 0.1014 x plus a 2 is 0.0007 x square.

So, this is our model, so for any value so, for example, for 25 we do not have a value of y right. So, we can plug in 25 over here and we will be we can calculate the value of y. So, the first value of stat is what we are interested at it is corresponds to r square. So, we can just say stats it will display all the four values or we can just say stats of ones s t a t s of 1 in the command window, if we had done this then it would only give 0.9874 right.

So, this is how we do polynomial regression, using the regress function right. Remember, we said like for polynomial regression there are two ways in MATLAB with which we can do one is using the regress function and the other one is using poly fit function. What we saw now was the regress function, we can now look into the poly fit function right.

(Refer Slide Time: 30:20)



So, this is the code that we currently discussed right and this is how what we are going to do using poly fit. So, the syntax for poly fit is x comma y comma n right. So, remember here it was independent, the dependent variable comma the independent variable. Here, it is the other way first we need to give the independent variable then the dependent variable and the order that we want to fit in right. So, here if you remember we were we are actually working with y is equal to a naught plus a 1 a 1 x plus a 2 x square right.

So, we want to fit a second order polynomial right. So, that is the order of the polynomial right. So, this clc clear close all you know. Here, we are defining just the data points right and since we want a second order polynomial we set n is equal to 2 again similarly we are plotting the data points right adding x label and y level and then we are using this function poly fit right. So, poly fit can return only the coefficients it does not give us an r square value like.

So, we are using this poly fit, we are saying x comma y comma n right. So, n has been defined here to be 2. So, for the same data we can also fit a third order polynomial right. So, for example, if we wanted to fit y is equal to a naught plus a 1 x plus a 2 x square plus a 3 x cube, all that we had to do was instead of n is equal to 2, we need to say n is equal to 3, remember unlike in regress where in we were forming this matrix x by stacking a column of ones and then having x values x square, we do not need to do that for poly fit function right.

So, we have this coefficients one thing you need to remember is the order in which we get the coefficient is the reverse as what we have what we would obtain in regress right. So, in regress the model was 34.2 plus. So, the first value was 34.2, because the once we are stacked the plus 0.1014 x plus 0.0007 x square right, over here b is arranged in the reverse order right. So, that you need to be careful about it right.

So, when you are interpreting this b values, the last term is the coefficient the last, but one term is the coefficient of x, the last, but second term is coefficient of x square and so on right. So, that would be b again this hold on is same thing we want to still visualize it. This exploit is also the same thing, y plot here we were actually constructing the model, because this was our model y is equal to a naught plus a 1 x plus a 2 x square right here we can directly use a function called as polyval right.

So, polyval the input to the polyval is the coefficients of the polynomial which is actually in BETA and the points at which we want to find the y values right. So, x plot is the 1000 points which we generated. Remember, between 10 and 60 we are generating 1000 points right linearly spaced points. So, those points are in x plot. So, the syntax for poly values give the model coefficients followed by the points at which you want the value of the dependent variable right.

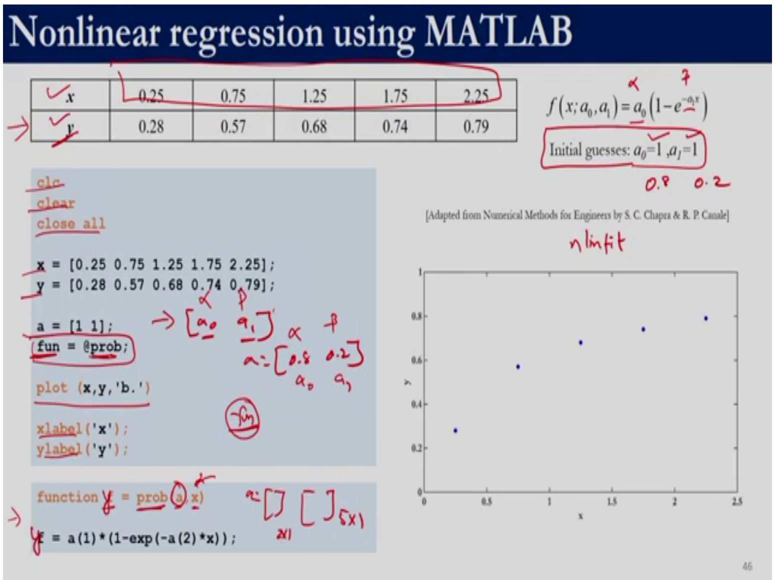
So, this will give y plot right again, plot x come x plot comma y plot again, we should have actually done comma r dot this right and then we add this legend measured comma model. So, if you run this you would so, if you execute this program you will still get the same plot which we got in regression right.

So, the good thing about polyfit is we do not need to compose this, x matrix another feature of polyfit is that you can directly use this polyval to plot the y values again, you do not need to explicitly state what is the model as we did in regression right, but to the best of my knowledge polyfit is does not give us the r square value right whereas, here we are using regress we can also get the r square value.

So, the choice is yours I mean depending upon the situation you can either choose to use the regress function or you can choose to use the poly fit function right. So, so far what we have seen is how to use a regress function for simple linear regression, for multi linear regression, we saw how to use it in the presence of constant coefficient and in the absence of constant coefficient then for polynomial regulation.

We saw it can be done with for polynomial regression, we saw that it can be done using the regress function as well as it can be done using the polyfit and polyval functions right. So, in all these cases we are working only with linear models even if the model was non-linear we transformed remember the couple of problems where in we had lawn function to transform it into linear form right. So, now, we look into the function n lin fit right. So, that can be used to directly solve a non-linear regression problem all right.

(Refer Slide Time: 35:33)



So, in this case we have been given this data x and y right and we have been asked to fit this model a naught into 1 minus e power minus a 1 x right. So, this nl fit also requires initial starting point. So, just like for all non-linear equations you need a starting point, for non-linear regression also we need a starting values right. So, a naught is equal to 1 a 1 is equal to 1. So, this is what we are going to start with right.

So, these are not optimal values right. So, that is the task of the function n lin fit, to come up with better values of a naught and a 1 right. So, the way to go the way to do it is first we need to define the data points before that clc clear close all as usual right. So, this is the case that we are going to work with a is equal to 1 comma 1, 1 comma 1 we have chosen the convention to be a naught a 1 right.

So, wherever we get the values, the first value will correspond to a naught the second value will correspond to 1. So, if the initial guess instead of 1 1 had to had it been 0.8 and 0.2, then we would have given instead of 1 1 we would have given a is equal to 0.8 and 0.2.

So, it is important to have a convention and stick to it throughout the problem. So, we will stick with this a naught comma a 1. We will define a variable fun right it is actually a function handle. So, we have a function whose name is prop right. We look into that function right that function, we have this at the rate symbol before this prop right. So, this basically means that we have a function handle whose name is fun right.

So, whenever we access use this variable fun, we are actually referring to this function prob right prob. It will receive two variables a and x right and it will return f. So, what it basically means that if you give a naught value and a 1 value in a right and if you give this x points it will give y not this y, but the y as per the model right.

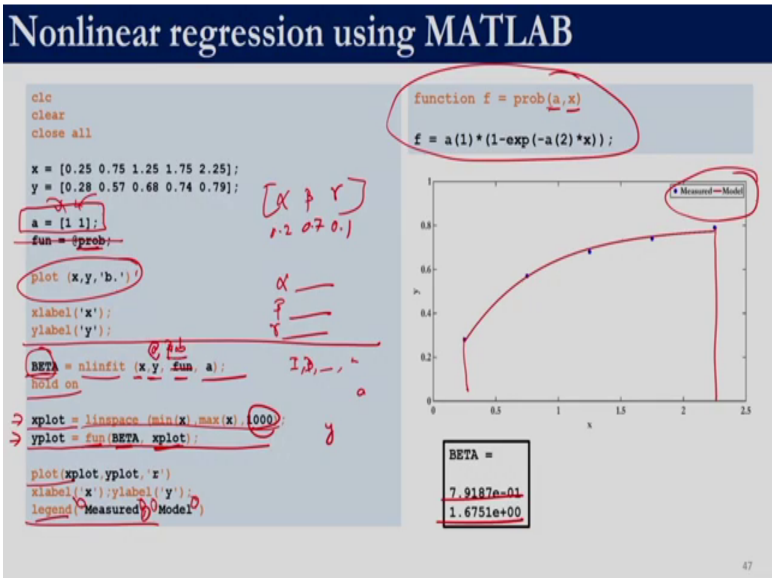
So, that is what we are writing here. So, the model is we could have chosen y and y over here right. So, y is equal to a naught right, but MATLAB the index in MATLAB will starts from 1. So, a of 1 right a of 1, because that is the convention that we have chosen. So, first value is a naught. So, a naught into 1 minus exponential minus a 1 x right, a 1 x here, we are writing a of 2, because the second value actually corresponds to a 1 right.

So, if all this is if all this is confusing you can just work with alpha and BETA right. So, here also it is alpha BETA alpha BETA right. So, the first value is alpha, the second value is BETA right. So, that way we can calculate y. So, the purpose of this function is given any values for a naught and a 1 and the data points it will tell you what is the value of the dependent variable this function will tell. So, x in this case it is 5 data points right.

So, x is going to be a 5 cross 1 right and a is going to be a 2 cross 1, because we have two parameters right. So, after that, so that is how this function is going to help us right. So, when n lin fit is going to solve this problem it is going to repeatedly call this function multiple times. You can see that by removing the semicolon and solving the problem right, you will get a y being printed repeatedly in the command window ok.

So, coming back to our problem right; so, here we are just merely plotting the data points and just giving the labels right, x label and y label. So, far remember we have not started solving regression problem, we have just defined whatever is required.

(Refer Slide Time: 39:44)



So we have discussed so far right. So, it is the same thing that we have discussed right. So, in the index for n lin fit is right. So, the independent variable followed by the dependent variable; followed by a function handle right. So, here it is fun, because the problem that we have is actually in returning the function fun right. So, we wrote it in prob, we have assigned that to fun right. So, the independent variable, dependent variable followed by if the function where we have the problem returned.

So, this is the problem that we have comma the initial guess right. So, if you had 5 variables let us say if you had a problem involving 5 parameters which have to be determined then a would be a 5 cross 1 right. So, the output of n lin fit we are receiving in a variable called as BETA right, you can give any variable name over there right and we also want to plot the model, in addition to the points we want to plot the model.

So, we have this hold on again as usual from the minimum value of x to the maximum value of x, we are generating 1000 linearly spaced points right and then what we are doing is so, that will give us x plot right. So, the BETA will have the model coefficients right model coefficients, it will have as many values as many values are there in this initial guess right, if there had been 3 values BETA will be a 3 cross 1 vector right, the first value will correspond to this first variable, the second value will correspond to this second variable.

So, if we had let us say three variable alpha BETA gamma that is how the guess we gave let us say 0.27 and 0.1 then BETA will have 3 values. The first value will correspond to the optimized value of alpha, second value will correspond to optimized value of BETA and the third value will correspond to optimized value of gamma right.

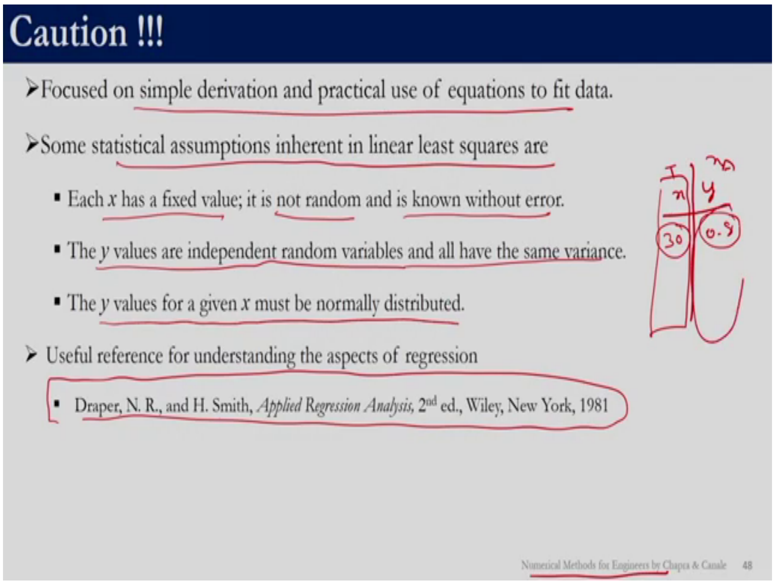
So, once we have this now, we need to find out what is the coefficient what are the values of the dependent variable y if the model coefficients are given in BETA right. So, BETA if we type in command window we will get this model right. So, if you stop with this, you already have BETA you already have solved the problem as in like you have determined the coefficients right, but here we are also interested in plotting that is why we are discussing the rest of it right.

So, for these x values we need the y value right. So, that we can find out using by passing this BETA values, the model coefficient and the x values remember that is what this format is the model coefficients comma the x values right. So, if we give that we get this y y plot right. So, and then we can plot x plot comma y plot r dot we can give right.

So, it will plot 1000 points right, if you increase this number of points you will get a smoother a smooth curve right and then we add the legend right, because the first we had initially plotted the points and then the model we write measured comma model again, within single quotes separated by a comma using the legend function.

So, it will help us to plot this plot right. So, that is how. So, that is how we can use the n lin fit function to do non-linear regression. So, again remember if we are we directly going to use prob over here then we need to have the at the rate symbol over here right. So, before concluding a word of caution right so, this is has taken from numerical methods for engineers by Chapra and Canale.

(Refer Slide Time: 43:35)



So, what we have been focusing in this part of the courses a simple derivation and practical use of equations to fit our data right. However, there are lot of statistical assumptions which are inherent in linear least squares which we have not discussed in detail if you are interested you can look into this reference right the book by Draper and Smith right.

So, some of the assumptions involved are each x has a fixed value, it is not random and is known without error, only under this condition can we do linear regression right. So, the so, for example, x and y we said. So, if let us say if x is temperature and y is let us say some conversion right. So, when we write 30 comma 0.8 what we actually mean is that this 30 is known precisely that there is no error in this 30 right. There could be some error in this 0.8 which is what we are trying to find using linear regression model which is why we are doing regression that we want to find a model which best fits the data.

So, there is there should not be any error in this x and the y values are independent random variables and all of them have same variance. Similarly, the y values for a given x must be normally distributed. So, this y should be normally distributed. For the purpose of this course right whatever we have covered is sufficient right and most classical texts stop over here, but there are also other measures right which you can if you are interested in regression, you can look into that as and when required right.

So, here we have a limited ourselves to the simple derivation of linear regression and how to use it provided the data satisfies all the assumptions that are required for linear regression right. These are some of the references that you can look into this book by Chapra and Canale there is also applied numerical methods with MATLAB for engineers and scientists by Chapra right. So, here you will find a lot of MATLAB related functions right, there is also this book Matthews and Finks.

Particularly, if you are looking for the derivation of the general linear least squares the transpose that into a is equal to z transpose y right. So, that derivation is actually given in this book right and for a advanced concepts of regression you can look into this Draper and Smith right. So, in this session we have seen what is regression we saw what is linear regression. Linear regression we discussed three types linear: simple linear regression, multi linear regression and polynomial regression, all these three with model with constant coefficient and without the constant coefficient.

Then we looked into some data transformation techniques we subsequently saw how do we solve a non-linear regression problem and then in MATLAB we learnt about four functions right polyfit, polyval, regress and n lin fit. So, regress is to be used for linear regression right, polyfit and polyval is to be used for polynomial fitting, where as n lean fit is to be used for non-linear regression right. So, with that we will conclude this session.

Thank you.