Course material

Course:

ENG606 / PHYS 442

Video:

DOE_lesson6_part2_fitlm_routine

Concepts (extracted from automatically generated subtitles):

Model matrix. Fit routine. Linear models. Full advantage of a routine. Typical data. Check of the leverage. Interesting classes. Different matrices. Main fit routine. P values. Case of cocktail. First part of the model. Rest of my model. Very useful function. Adjusted s square.

to video sequence search
(within ENG606 / PHYS 442.)

to video

...

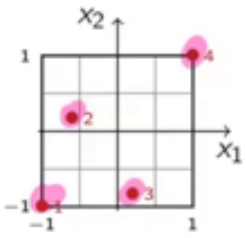
notes

summary

0m 0s

3.3.7 Let's go back to the case of the cocktail

Runs



Model matrix

$$X = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & -0.6 & 0.17 & -0.1 \\ 1 & 0.2 & -0.83 & -0.17 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -0.6 & 0.17 & -0.1 \\ 1 & 0.2 & -0.83 & -0.17 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Corrected sum of squares

$a_o | a_1, a_2, a_{12} \rightarrow A_o = (-0.1 \quad -0.165 \; 0.433)$

which means that :

$a_o^* = a_o - 0.1a_1 - 0.165a_2 + 0.433a_{12}$

Then :

$SS(a_o | a_1, a_2, a_{12}) =$

$$\left(a_o + A_o \begin{bmatrix} a_1 \\ a_2 \\ a_{12} \end{bmatrix} \right)^T X_o^T X_o \left(a_0 + A_o \begin{bmatrix} a_1 \\ a_2 \\ a_{12} \end{bmatrix} \right)$$

And so on for the next steps :

$$a_o, a_1 | a_2, a_{12} \rightarrow A_1 = \begin{pmatrix} -0.09 & 0.44 \\ 0.71 & 0.09 \end{pmatrix}$$

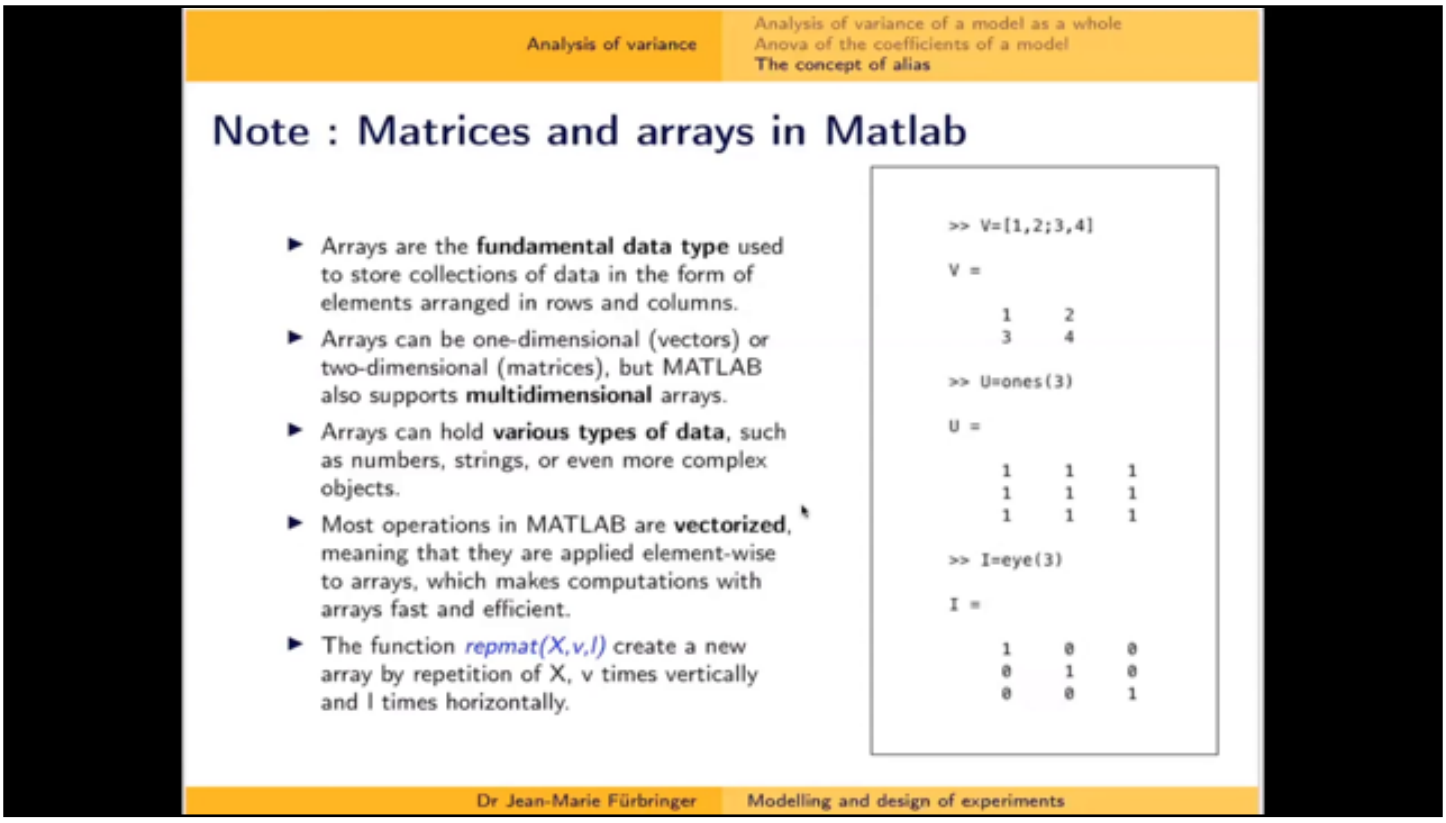
$$a_o, a_1 | a_2, a_{12} \rightarrow A_2 = \begin{pmatrix} 0.46 \\ -0.04 \\ 0.18 \end{pmatrix}$$

These subtitles have been generated automatically I'm back to my case of cocktail and I just have put in red the position of the points.

notes

summary

0m 1s

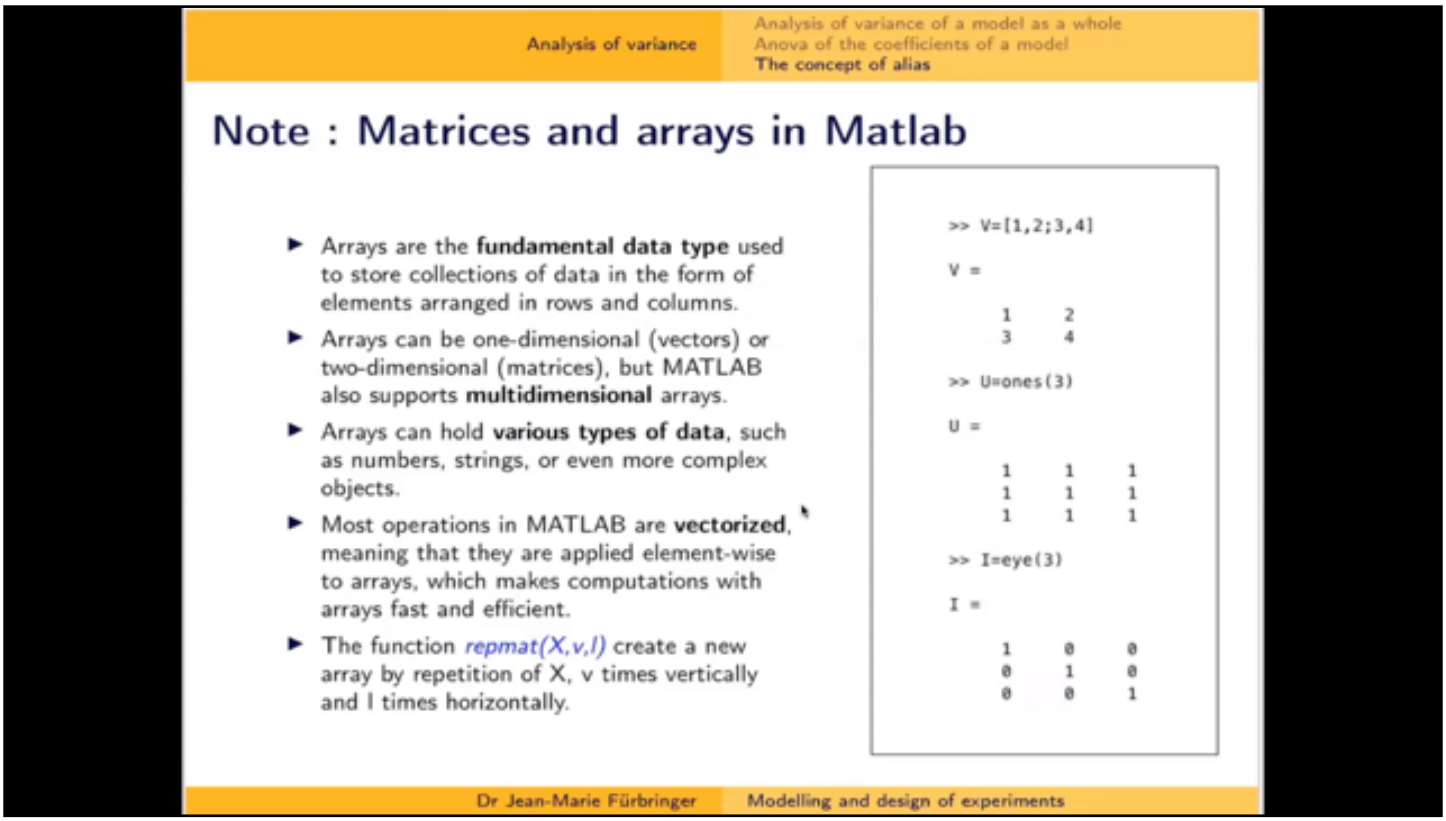


I see, I don't know if you see it, when I just see where are the points, I understand it's not orthogonal. After this time, you will guess it's not orthogonal, but I already see. So I'm able to have my model matrix for this case. You remember one column for the constant, 111, the first and the second column corresponding to my factor x1 and x2. Be careful, I'm using x1, x2 for the factors and sometimes for the different matrices. And here is the interaction, the column of the interaction, which is the product of the second and third column of this matrix is how I built my model matrix. So I'm interested to calculate the corrected sum of square. I understand that all this I'm doing is for calculating a nanova, which is correct and orthogonalized things. So I'm interesting to calculate my a0, the residue and my sum of square of a0 knowing a1, a2, a3. And so I have calculated my alias matrix, what I call a big a0 is the alias matrix of the case when I have on one side the constant and the other side the rest of my model. So in this case, my alias matrix is the raw matrix because I have one coefficient in the first part of the model and three coefficients in the other part of the model. And then that means that my corrected coefficient, the one I will use for calculating my sum, my corrected sum of square, that will be a0 minus 10% of a1 minus 16% of a2 and plus 43% of a12. And then I can calculate my sum of square of the corrected coefficient. The corrected sum of square will be sum of square of a0 knowing a1, a2, a12. So this would be obtained making the calculation. You see on both part is the sum of square. So here's the

notes

summary

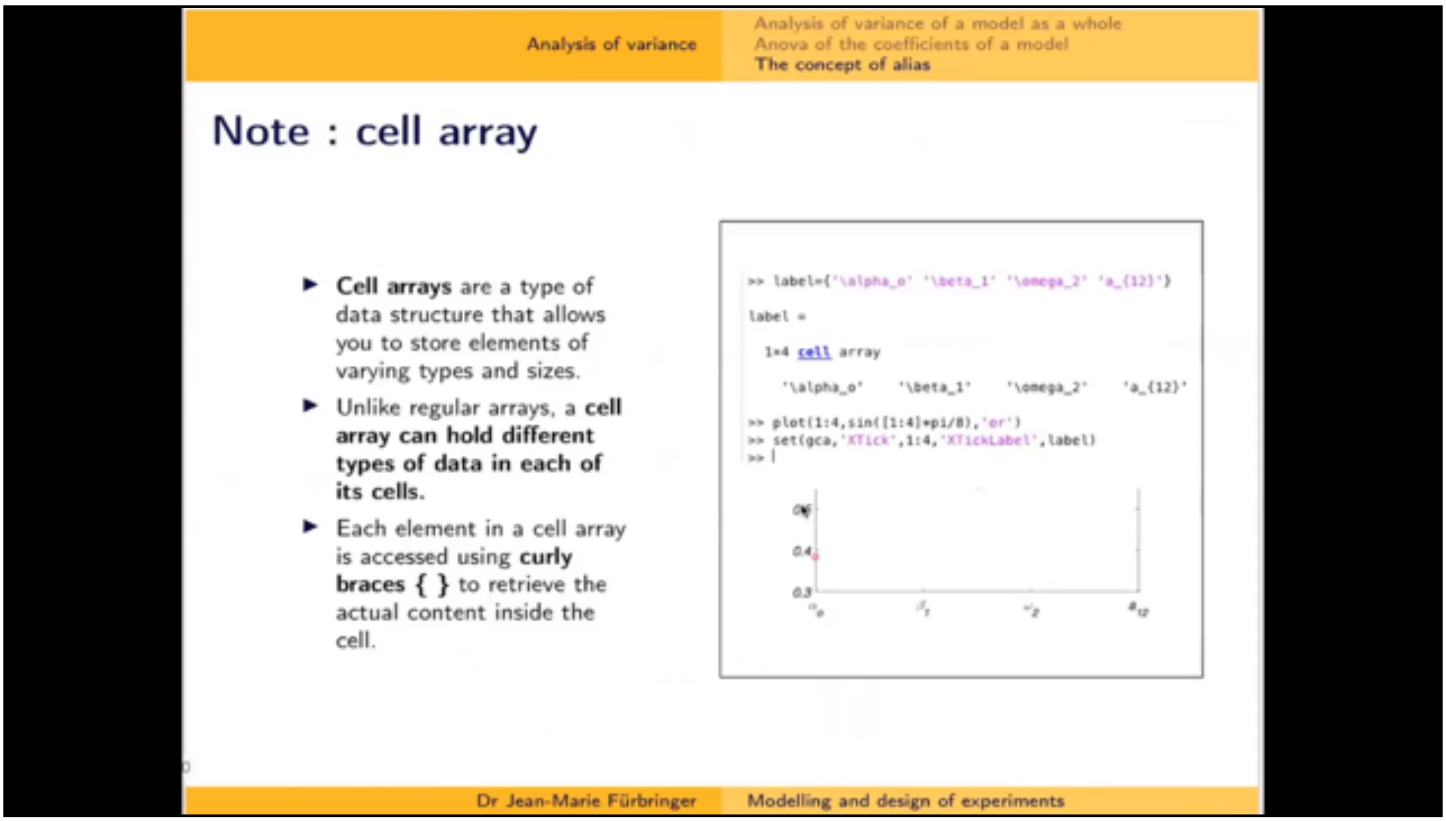
0m 5s



square of coefficient. So x0 multiplied by a0. So I'm reading here a0 plus big a0 multiplied by the vector a1 and 2. And like that, we can also do the other calculation. And after we can have a0, a1 calculating a2. Okay, I will not finish reading all this. You understand how it works. So it's an application of the alias. Thanks. Some years I go through this very rapidly and the impression that you cannot understand. And now I explain everything in detail. I don't know. I'm not sure it's so understandable. So very sorry. Just wanted to make the case complete. And you understand that it's where things are coming on and how we can calculate. Again, it's not the most critical part of my course. And I present to you a few elements of MATLAB that are useful for taking a full advantage

notes

summary

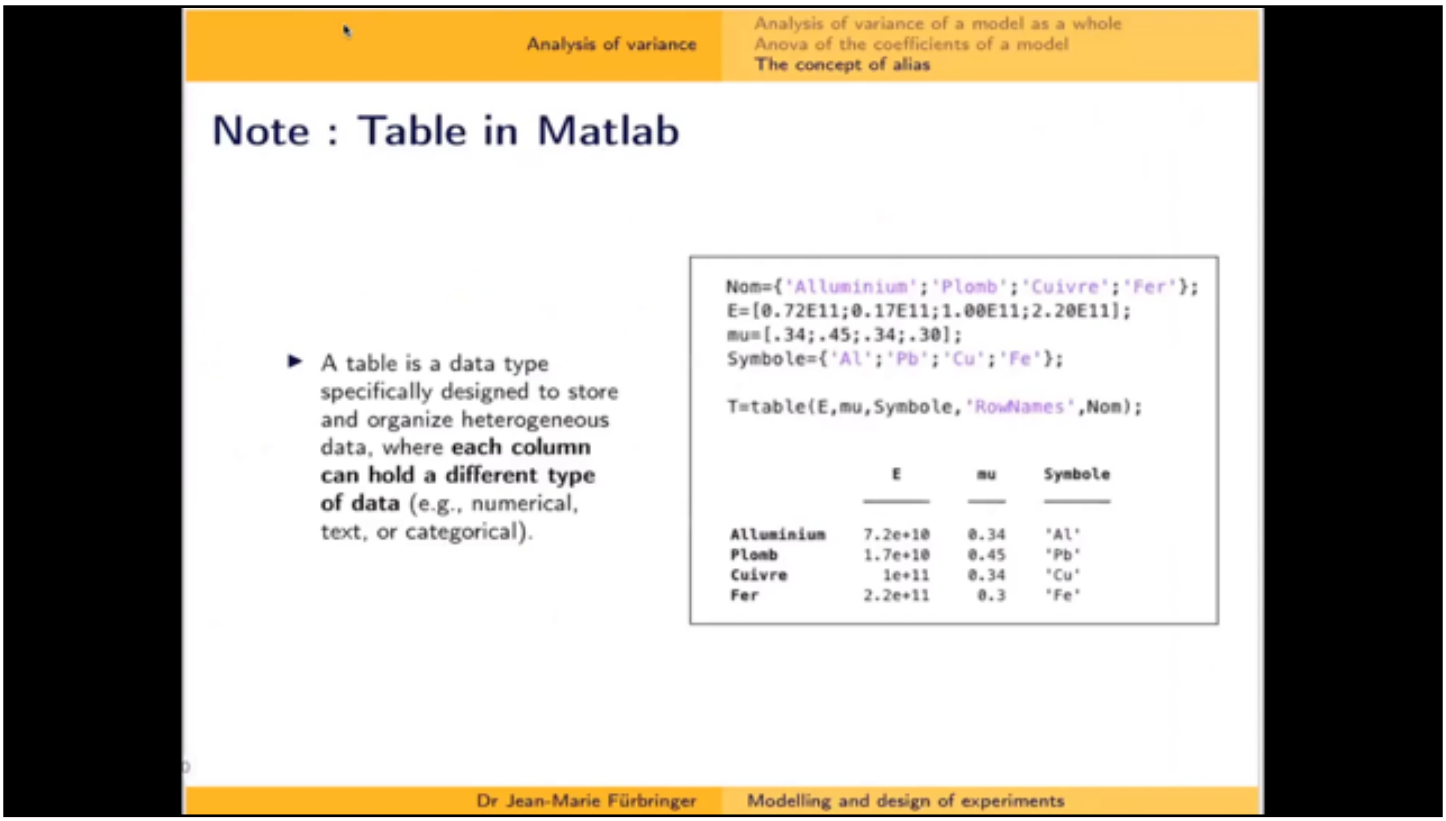


of a routine, which is called fitLM, which will be the fit routine, the main fit routine we will use. So in MATLAB, you have matrices of arrays. So I can pass rapidly these slides. If you make a little bit of MATLAB, you understand you have the same concept in Python. So it's OK. So here you have a few explanations. There are perhaps one function in the related to the arrays which is interested. This is function repMat that lets you copy a matrix horizontally and vertically, a number of times or make a sort of mosaic. And it's very useful because when we are playing with replicates of experiments, so we need to replicate the matrix of essay or the matrix of the model. So it's necessary. So remember repMat, write it somewhere for remembering because it's a very useful function. So this is very typical data that we use in MATLAB.

notes

summary

4m 1s

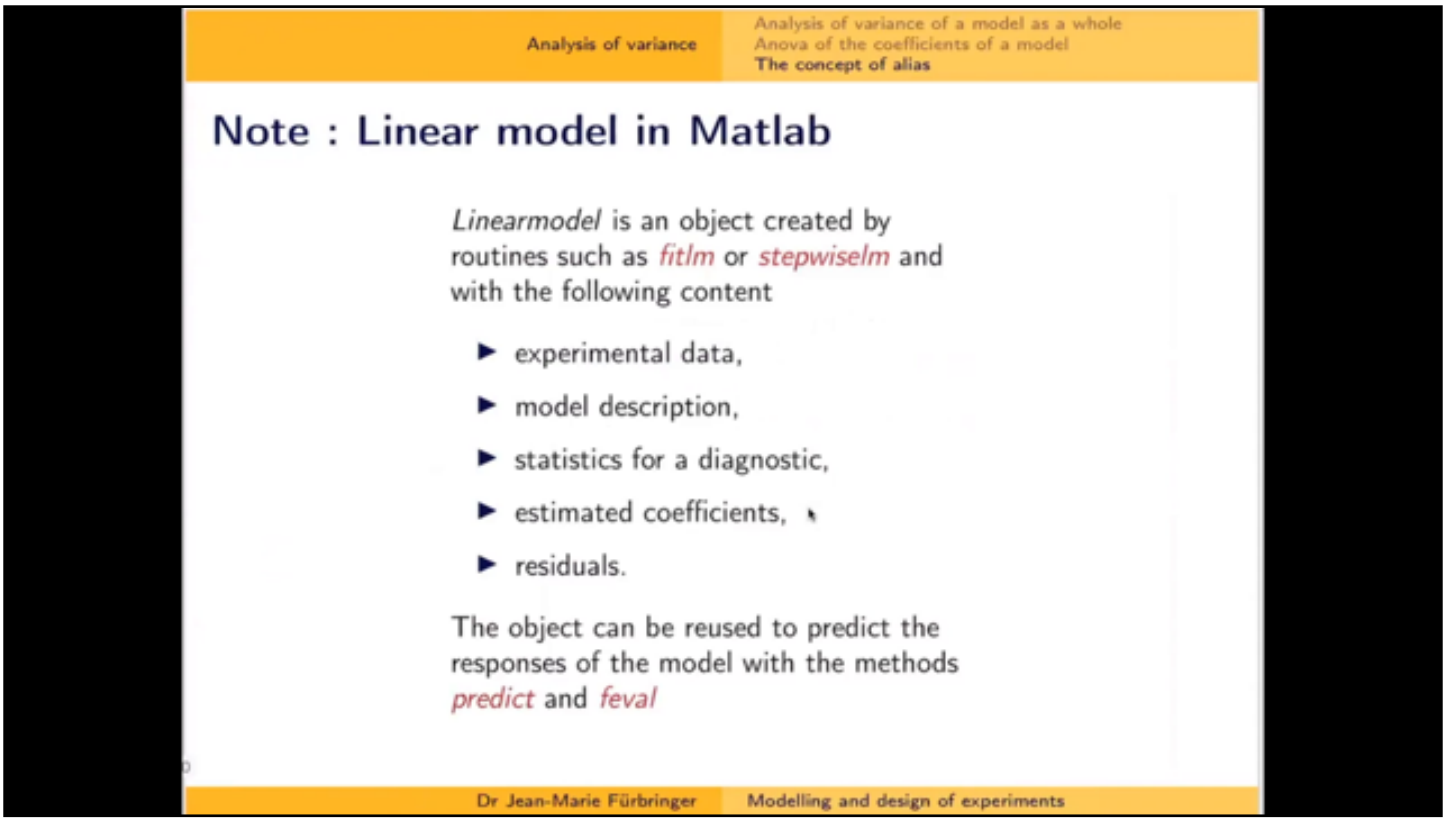


OK, so another interesting classes is cell array. So it lets you make a collection of objects interesting in the cell array. Then you can put together elements that are different. So you can have string, you can have a matrix, you can have matrix of different arrays of different dimension, and you can get them in one place. So very useful. So I mainly use the cell arrays for playing with the label ticks and also seeing this very useful for defining variable, having the correct name of your variables, etc. You see a small example where I put it. And when you use the algorithm, the function of MATLAB for text, you understand Latinx. So typically you see here for the alpha zero, backslash alpha underscore zero, it's make a beautiful alpha zero you see here in the... OK, so some tricks about MATLAB.

notes

summary

5m 8s

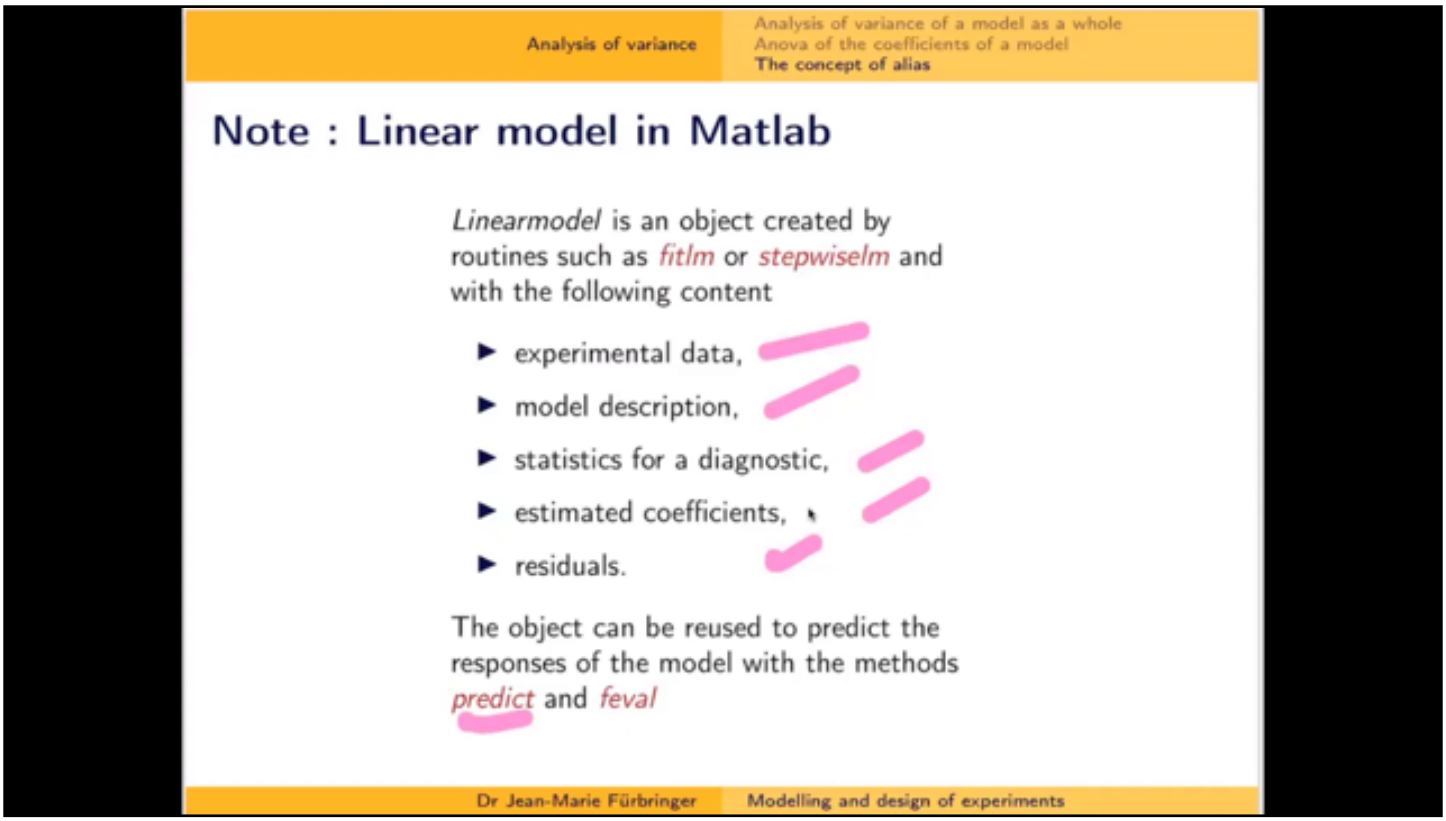


Something that I find also very interesting, I think already presented in the first chapter, but it's the tables, the interest of table that you keep the name of each column and very be careful with your matrices, with your arrays, because sometimes you don't remember which column was what and you have trouble with that. Also, the interest of table that you really I recommend you to use table. It's when you can, it's really a very interesting element.

notes

summary

6m 24s

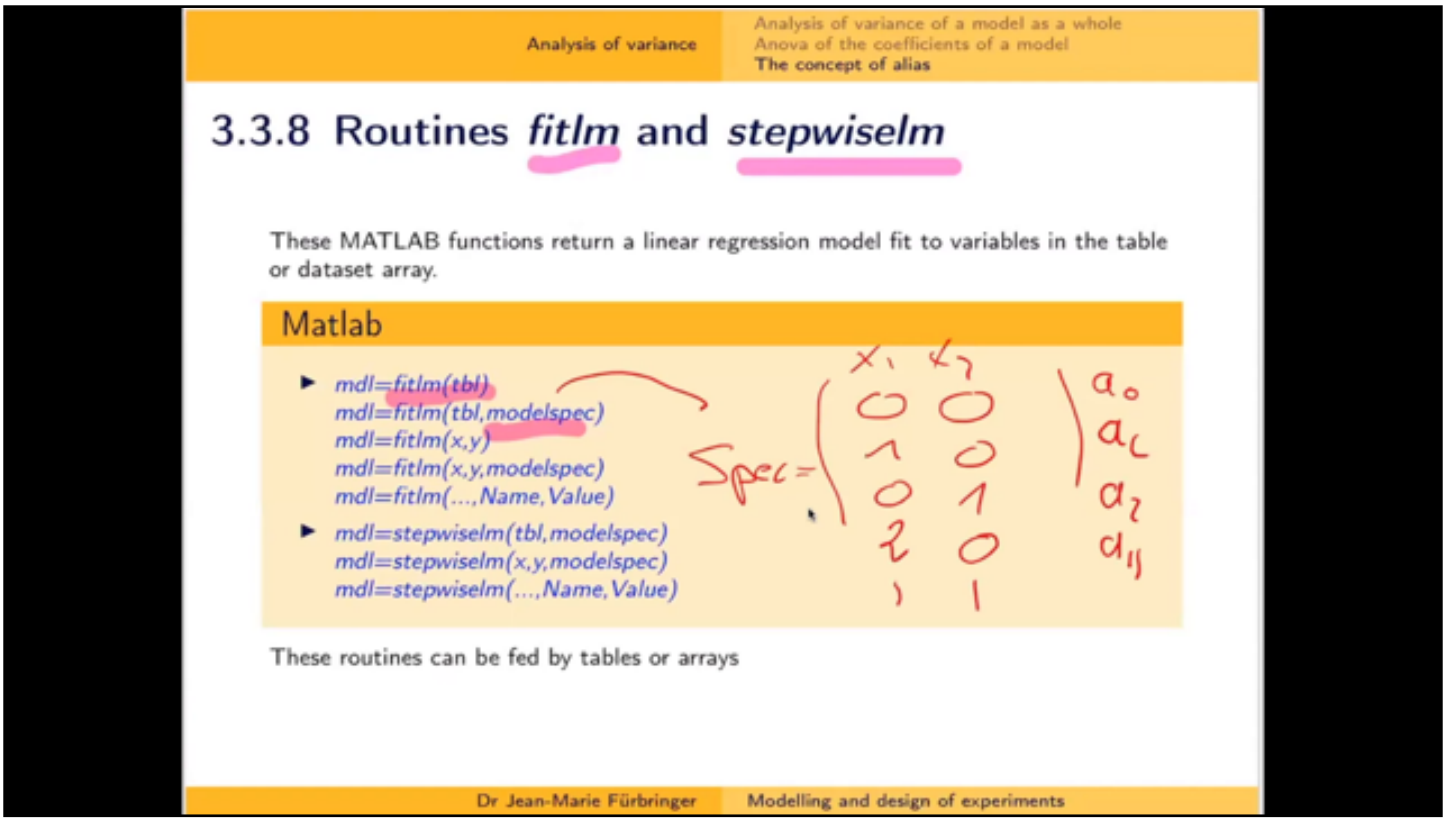


OK, and you have another type of classes which you know interest is linear models. So linear models are produced by routines like fitLM, the routine we will see and other type of routine of fit. And it's very useful because it makes all the calculation about the fit and after you have as a small database and you can use it. So you will have your experimental data, the model descriptions, the statistic and the diagnosis, you have the coefficient and you have the residual. So it's a way it's worked with MATLAB. You make a fit, you keep all your data and after you exploit your data for making the graphic, making the analysis and understanding what's what you have done. And you can reuse those type of model for calculating the ANOVA table, for

notes

summary

6m 58s

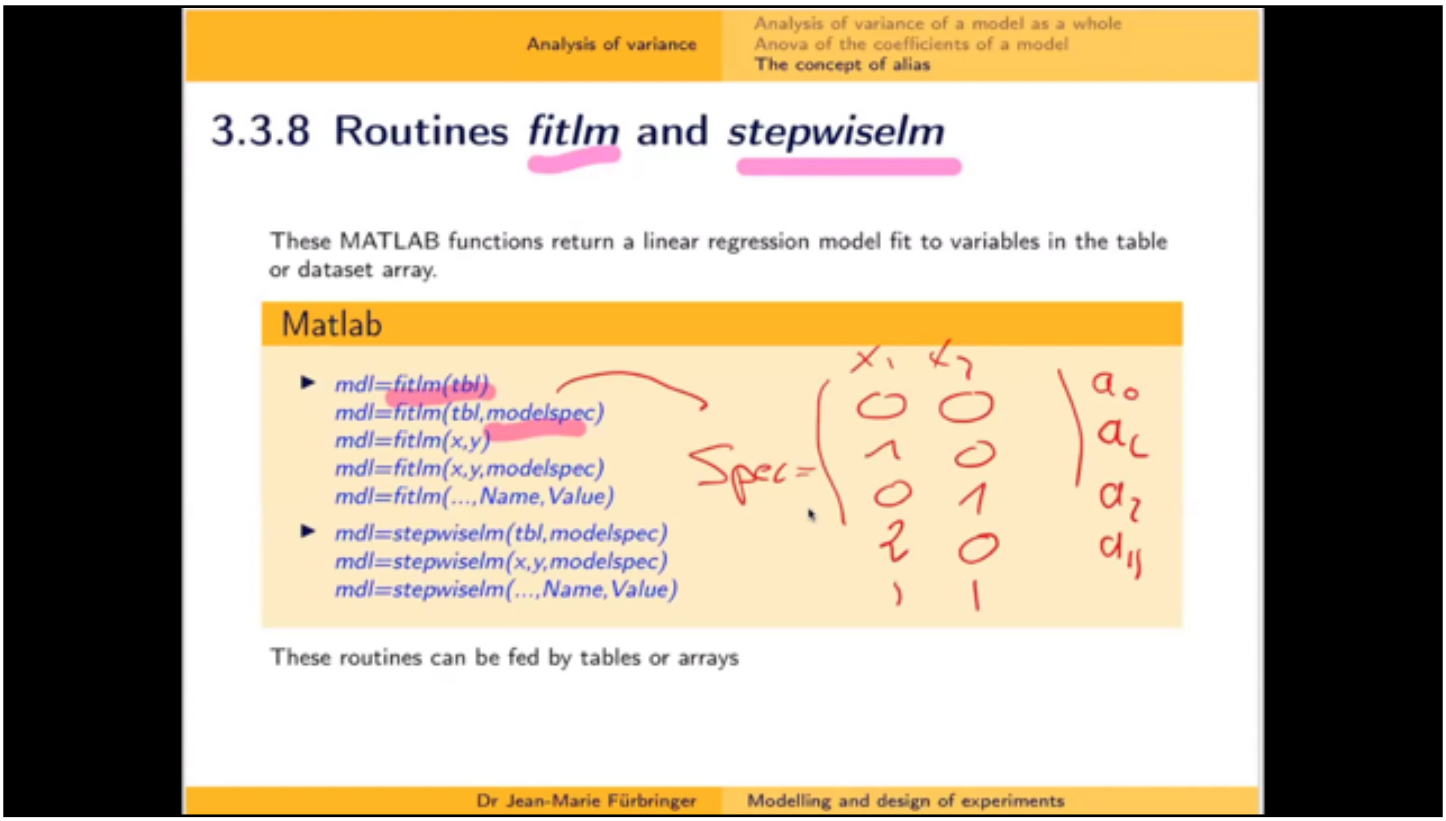


calculating, evaluate your data point where you want with function like predict and F able. OK, so all this is for presenting you the routine fitLM, LM for linear model fit a linear model and you have a function, a brother function, which is stepwise LM, which quite do the same thing, but step by step and you can put condition and the acceptability of your coefficient. So you can use it in different way. You can just submit or fit the function with a table and usually consider the last column as your answer. Everything that comes first as your model, your matrix of essay, your model, your matrix of experiment and the last one as the answer, the responses. If it's not the case, you can indicate you have also the possibility to tell which one is your data. So look in the help, you see a few details. Don't change your table. If you already have your table for the routine, you have a way also to tell which columns are what. So you can, if you do that, if you use just fitLM with just the data, the model, induced model is a linear model, just the constant and the main effect. You can also have a specification of the model and you have two ways of specify a model. To specify a model, you can have keywords like linear, interaction. I never know which one is with S and not S because MATLAB use the keyword interaction and sometimes the keyword interactions. I believe that in, and this one is with S and with X2FX is without an S. And you can also have model matrix, specification matrix, but you have a question? Yes. So the question was what means interaction for fitLM? It means linear plus interaction, but it's give you the interactions two by two. It doesn't go more away. So the specification

notes

summary

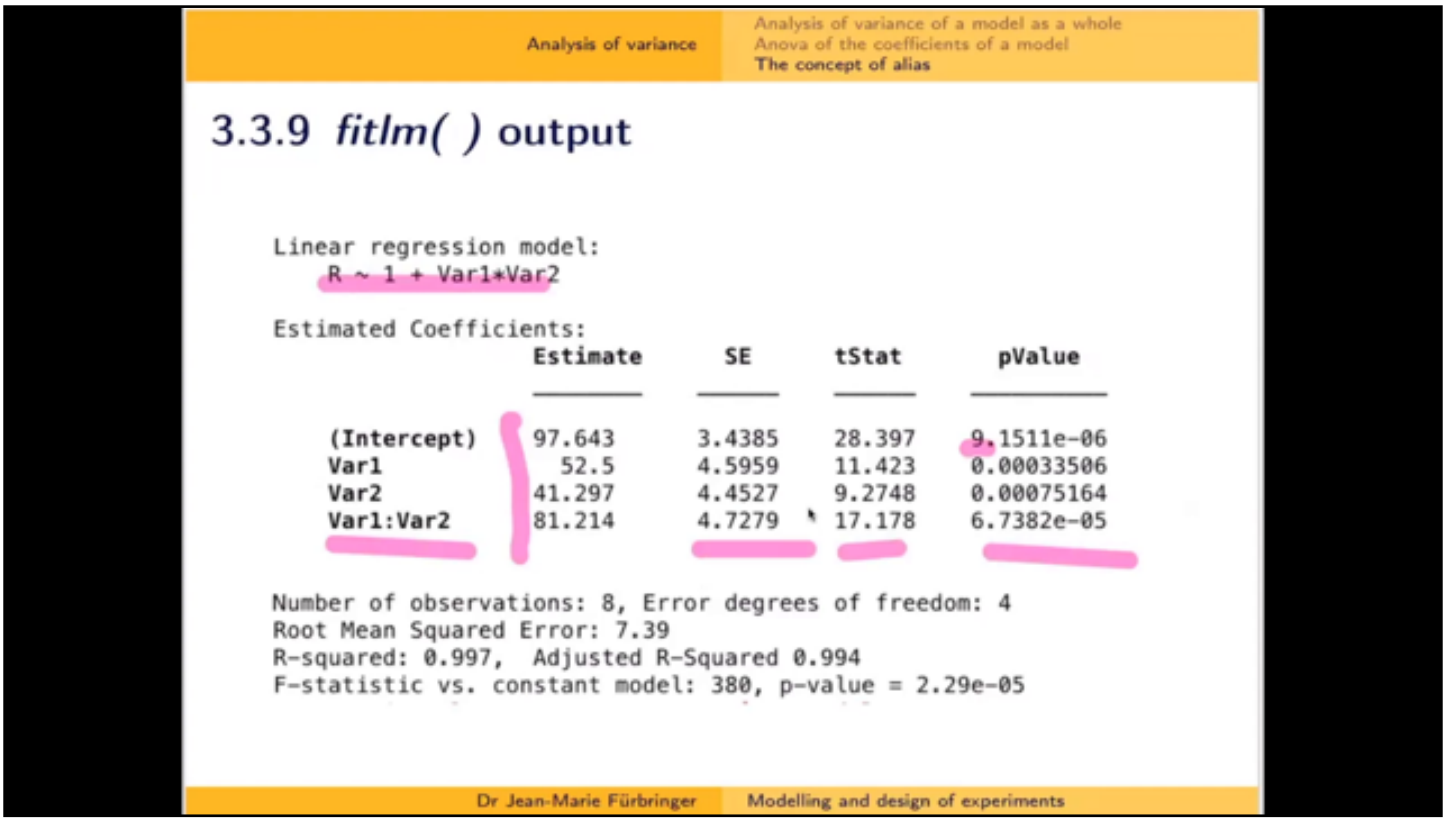
8m 1s



matrix could be, so for this, you can make, you can set a matrix called spec and it have as many columns as you have factors. So let's say you have X1 and X2. And after you indicate for each coefficient that you want in your model, you write something. So if you want the coefficient A0, so you indicate zero and zero because X1 is not implied in A0 and X2 is not implied in A0. If you want the coefficient A1, you write one zero. And if you want the coefficient A2, you write zero two because zero one. If you want, for example, so this was in two, if you want the quadratic A11, you write two zero. So this would be the square coefficient for A1. And if you want the interaction, you write one one. Like that, you can build all the possible linear model and you will

notes

summary

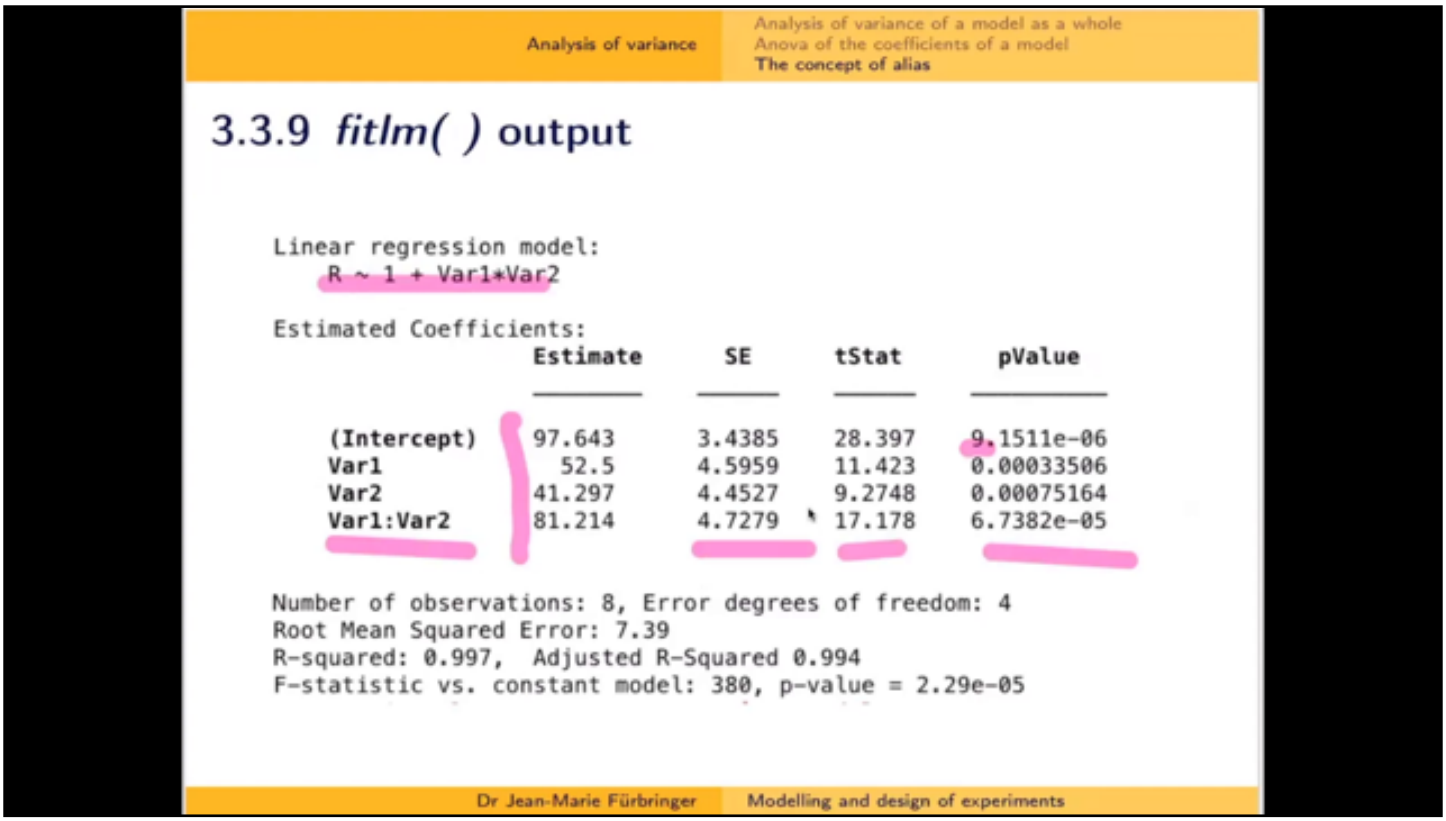


feed spec here at the place of keywords, linear, quadratic, etc. You can also fit the model with matrices with arrays. So you feed first with the matrix of essay. You don't give the matrix of the model. You write itself the matrix of the model. And why is your answer model spec the same thing? And after you can look in the help, you have a few elements that you can instruct. You can give an instruction to your algorithm, which is the names of your variables, the name of your output, and a few other elements that you can put. Stepwise work quite the same. And you can also indicate to stepwise your thresholds for accepting and introducing factors. Typically it's 5%, by default it's 5%, but you can change that and have. The interests of stepwise that you can give a very, very complex model and it will eliminate the coefficient one after the other that are not useful, that are not significant. This is the output of Fittelheim. He first tells you what he has fit. So I will make a slide on this explanation. This is the Wilkinson notation. So how he tells you that he was trying to fit R as a result. It was the name of the output variable. And he was trying to fit it with a constant for the one. And he tells you here that he, VAR1, multiplying or two star, VAR2, he tells you that they have considered the main effect of each one, and the interaction. We will see that in detail. But each time the algorithm tells you what you have fit as model. And after it gives you the estimate of the coefficient. So what he called intercept is a zero. And it gives you the main effect for the first variable, the second variable, and the interactions. So the element of interaction is VAR1,

notes

summary

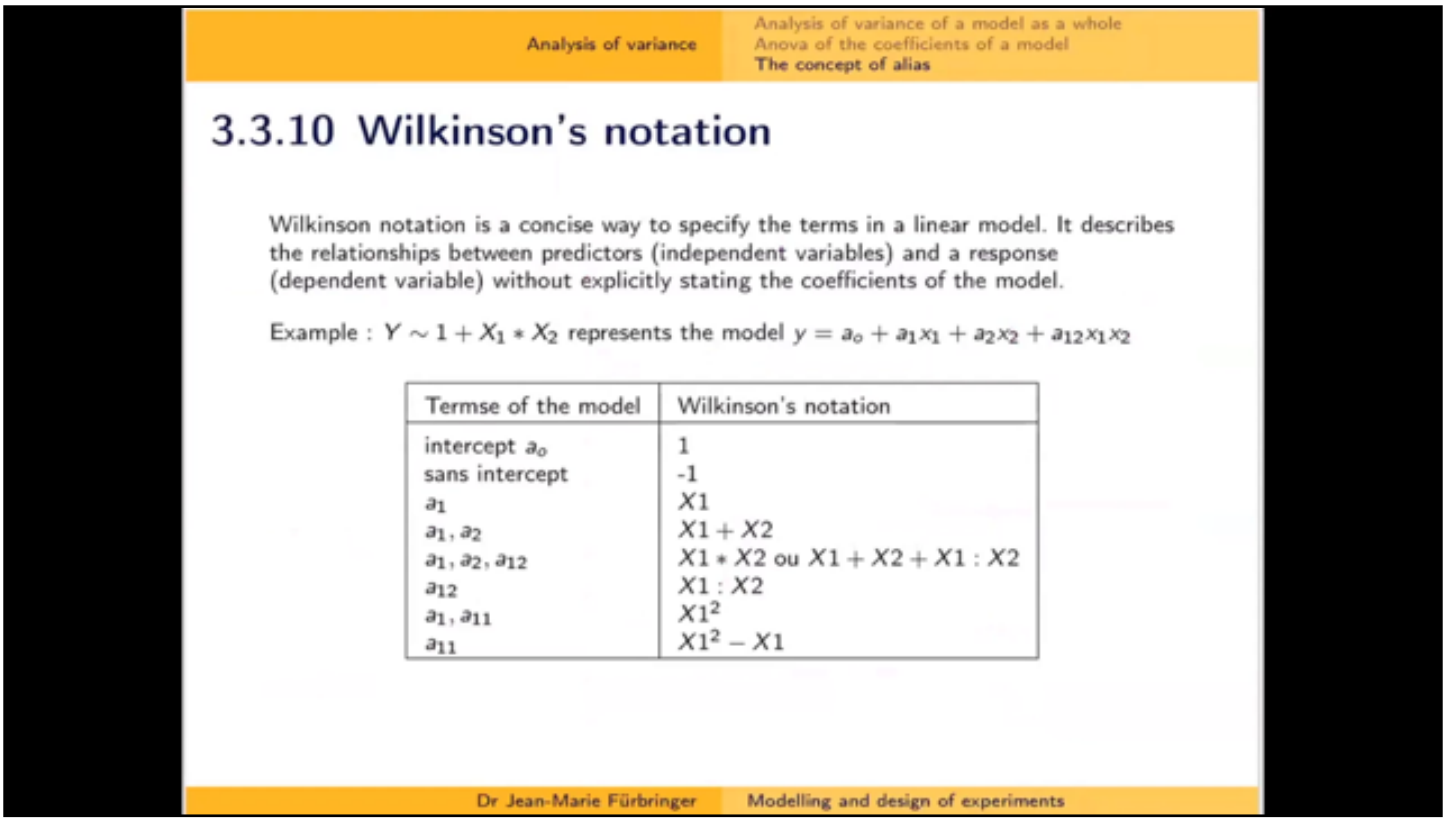
12m 1s



two dots, semicolumns, and VAR2. If we would have fed the names of the variable in the algorithm, it would have put there the name you have put. After it gives you a standard error of the coefficient. So it's very interesting because it's led to evaluate very rapidly if the coefficient plus minus is standard error is okay or is a zero. So we see that we have 9750 to 4181 plus minus something as four points something. So it seems that it's okay. So the result of the fit in this case seems to be quite okay. After it gives you the student statistics. You can understand what is a student statistic, but what I propose you jump to the p values. So in the p values, it's give you the probability that's what you have. Fet is distinct from the noise. So it's in fact the probability that if you accept this coefficient, you are making an error. So you see in this case, we are quite happy. They are all very small.

notes

summary

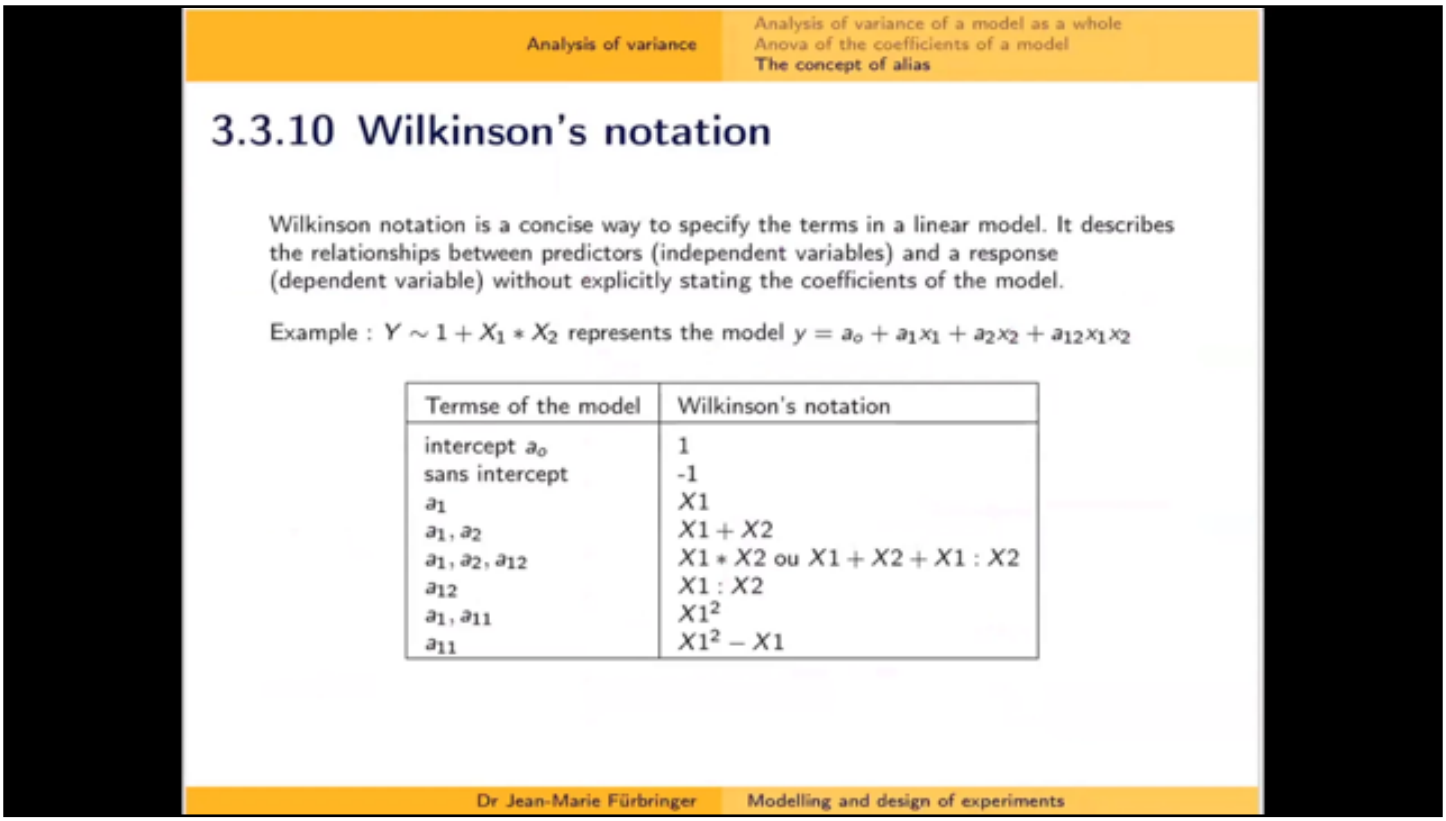


We have nine ppm, a percent per million because 10 power minus six is a ppm. So the probability to be wrong, accepting the constant is quite small. We have something as I don't know, three 10,000s for the first coefficient, seven 10,000s for the second variable and something as 67 ppm for the last. So the value of the p value of the different coefficient of my model are quite okay. They are smaller than five percent. Okay. So when you analyze that, the first thing that you have to check the value of the estimates, after you can check the standard error to see if your estimates are quite okay. And after you can look the p values and see if the values are acceptable. And as I mentioned, the standard thresholds for the p value is five percent, which is not very good. It's just acceptable. Usually you like to have something a lot smaller than five percent. But it depends also what you are doing. If you have a lot of noise or not a lot of noise. After you can check this part of the answer. So you see that here you can remember you that you have eight observations and you are estimating four coefficients. So you have four degrees of freedom for your error. It could be interesting to remember the root mean square error. So it's something comparable to the experimental error. The s square, I don't find it usually so much interesting. It's not a statistic. I appreciate a lot. But nevertheless, we say here that we are explaining 99 percent of the sum of square. We have a model which is very close to my measurement. But is more interested is the adjusted s square. 99 percent two, because here we can compare model with more or less coefficient. So the adjusted s square is really interested for comparing models

notes

summary

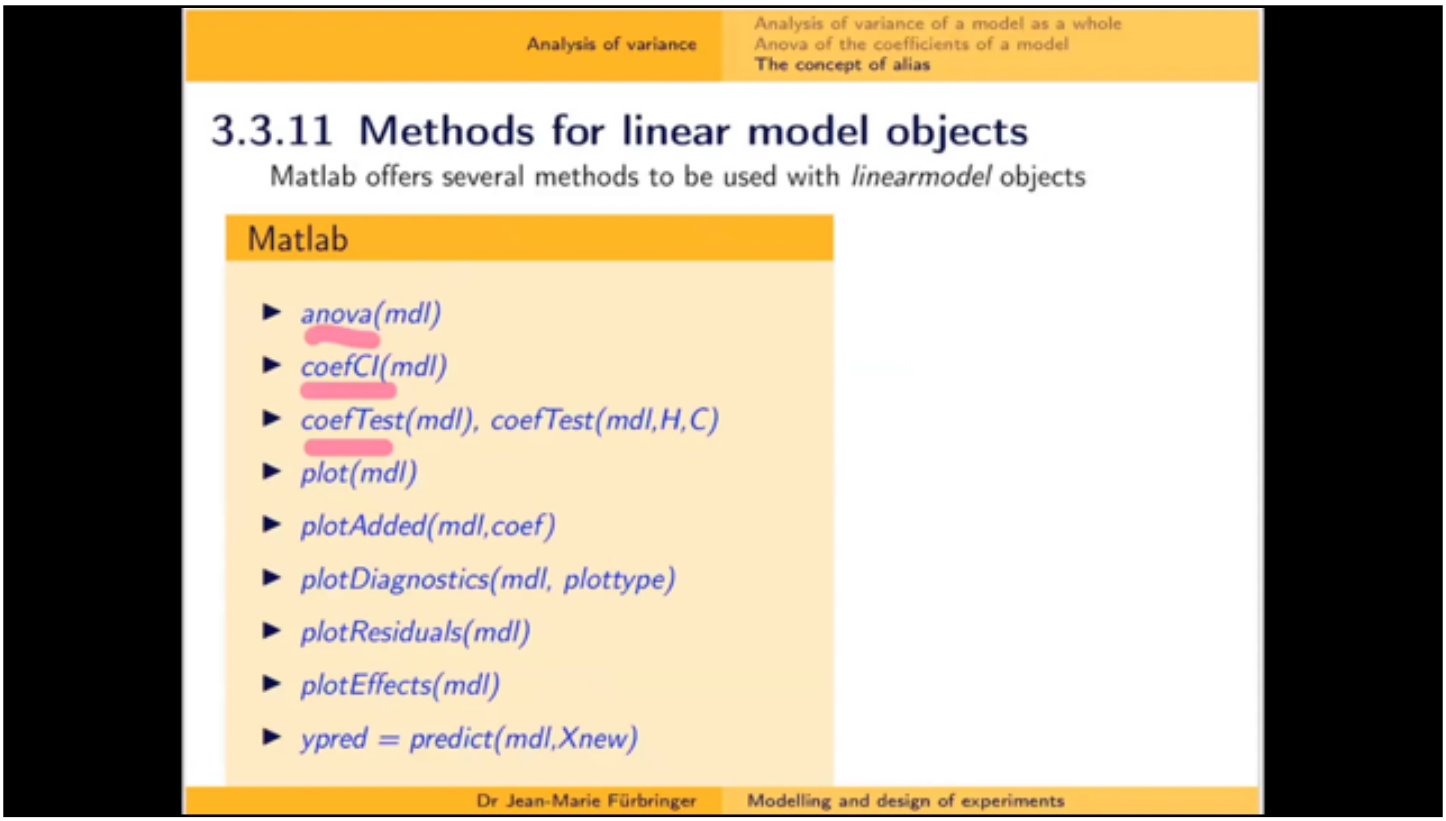
16m 1s



of the same on the same data, which do not have the same numbers of coefficient. You cannot compare just pure s square. You have to compare adjusted s square because evidently more you have coefficient, smaller is the residual all the time. But it doesn't mean that all the new coefficients you introduce in your model are really explaining a lot and that they are bigger than the residual. And after you have a p value for your model as a whole. So you see here that our model is quite OK is 22 ppm. It's quite a good model. So the evaluation here would be to say we have quite a good model and all the coefficients that we have considered are significant. OK, so this is the output. Understand that here all the test is based on student test. When we make a nanova is based on Fisher test. They are not the same, but the results are usually one to digit very close. It's never in contradiction. But the test for this is a student test trying to prove that our coefficients are different from the noise.

notes

summary

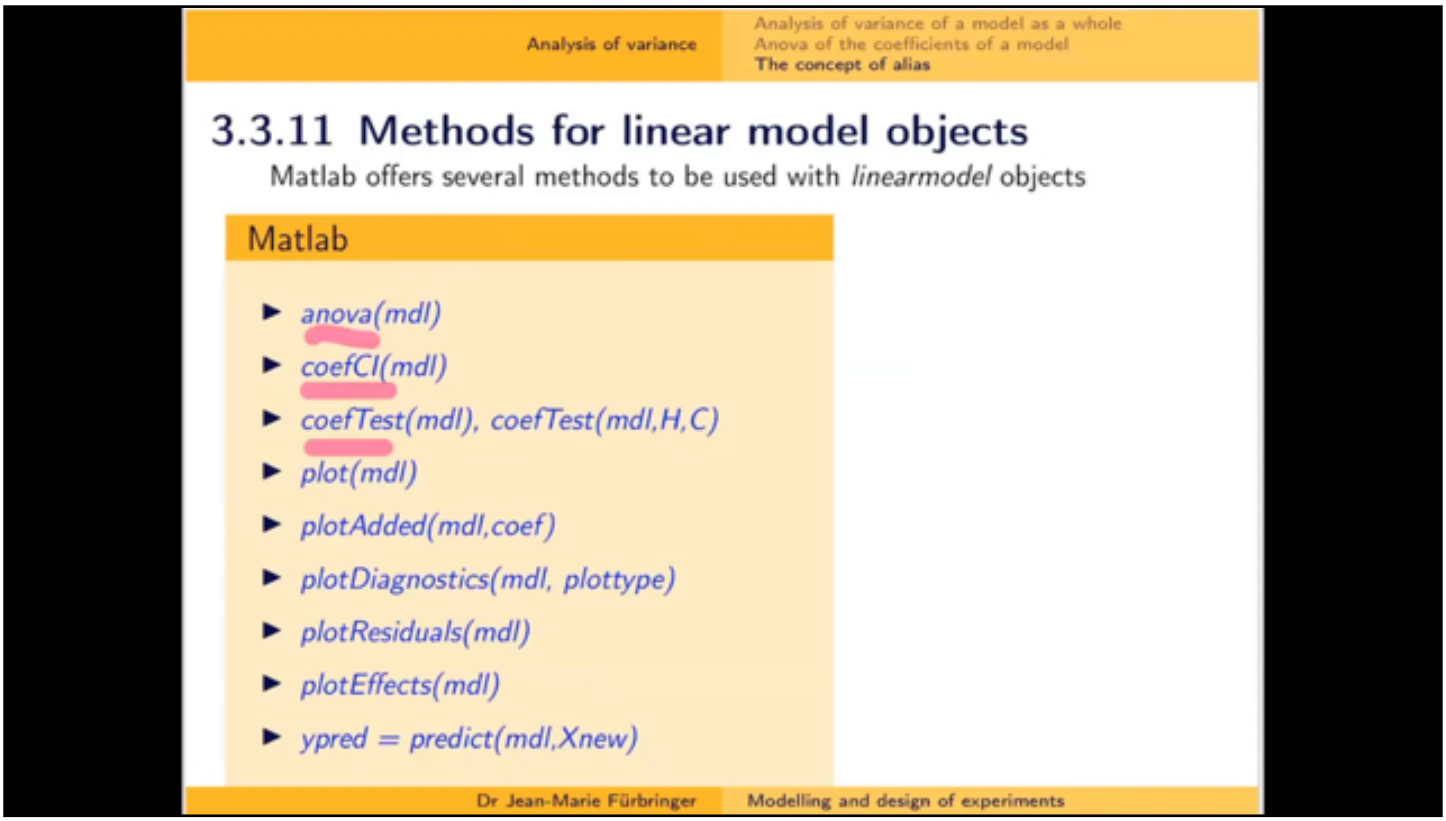


After I wanted to present this Wilkinson notation, some algorithm accept also this as a definition. So in the Wilkinson notation, we try to present this linear model as in the most simple way. So it's usually an output which is following this tilde is for saying following and we present the linear model. So one indicates that you have a constant minus one means that you don't have the constant. So when you put something that you indicate that you wanted in the model, when you put the negative sign, you indicate that you don't want it in the model. After if you want the linear model of the first factor, you put x1. If you want also the second factor, you put plus x2 plus x3 plus x4 for expressing that you want to integrate which factors you want to integrate in your model. After you can indicate the you can use the star sign for saying that you want the interaction and the main effects or you can indicate individually what you want and that you want also the interaction. So when you make semicolons, you indicate that you want the interaction by itself. When you put the multiplication, you indicate that you want also the main effect. And so you don't need to write x1 and x2. So it's a shorter notation. X1 multiplied by star x2 is equivalence to the notation x1 plus x2 plus x1 semicolonsx2. And if you just want to indicate some interactions so x1 semicolonsx2 indicates some instructions. So x1, semicolon x2, so you can select the different instructions that you want. If you want quadratic terms, you indicate that x1 square. In this case, you have automatically also the main effect. If you don't want the main effect, you have to write x1 square minus x1 for telling that you just want the quadratic terms. Wilkinson notation, I never did

notes

summary

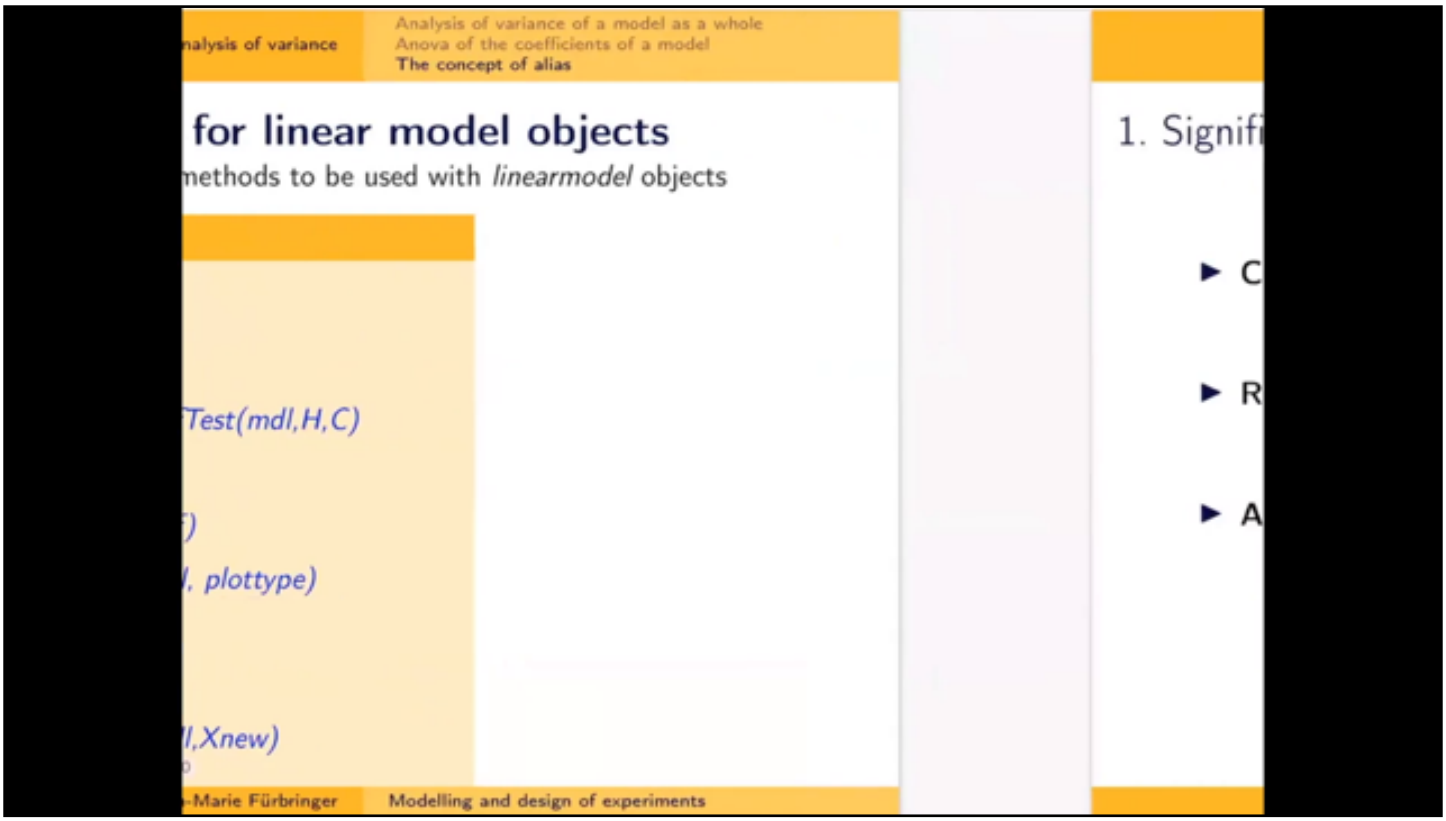
20m 1s



it, but you can also feed ptlm with this, but I never do it. I prefer my matrix of the model. I find it a lot more easy, but you can also do that. The last MATLAB is all the same answer. So I don't know in Python, in R, what is it? Eventually, you have one or the other notation with this fit lm. We have what we call method. So when you have a model, you can explode the model with the model that you have obtained. So you can calculate the ANOVA table. In this case, we will perform a Fisher test. You can calculate the interval confidence of the coefficient with cofce. You can calculate the test. So understand that the p-value is testing that your coefficient is different from zero, that your coefficient is different from the noise. If you want to test the real value, you need to add a new test. So you do that with cof test. Eventually, you obtain as estimate 2.1 and you say, okay, I want to consider 2 because it is your model. You would like to test what is the probability that this value 2 is interesting. Most of people don't do it, but

notes

summary

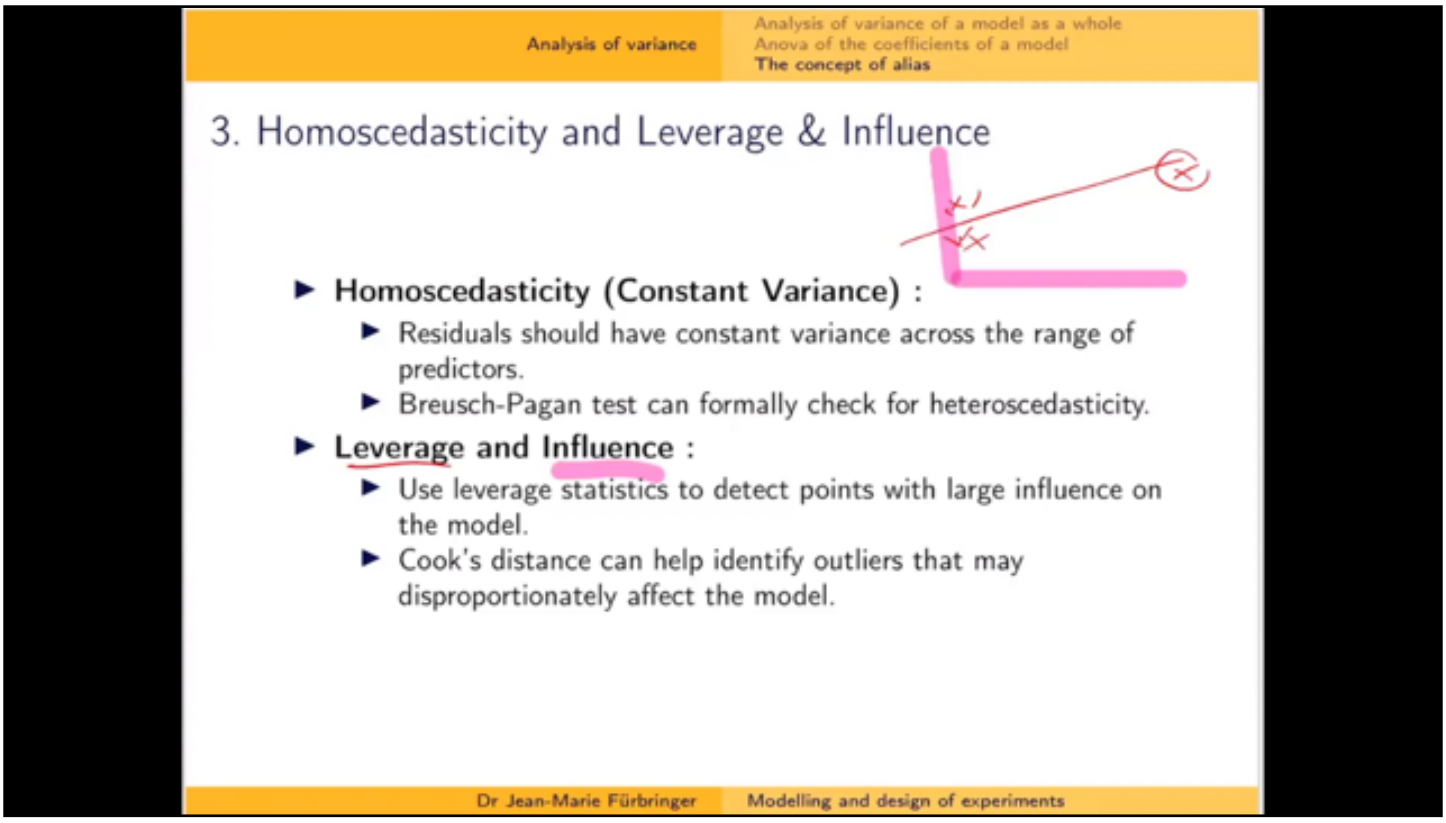


usually we should do it to make correct statistics. You can plot a model. Usually, it makes you plot very interesting the data against your estimate and you see if it's linear. Plot added, it's also very interesting. It lets you check factor after factor. What's each factor is adding to the model for testing when you want to consider or not. Diagnostics is for checking the residue. Residual is also doing quite the same. You can plot effects and you can make predictions. So I really don't want to enter in the L but take time and I will precise but I have one file where I have used all this function and I see you or you can use it. I recognize that the help of MATLAB is useful most of the case but sometimes it's quite complicated. But don't hesitate. It's a trick. Don't hesitate to use something as chat GPT you ask. Explain me this routine or usually it gives you a comprehensive explanation on elements that perhaps

notes

summary

24m 1s



is not so easy. After you have a few slides explaining step by step what is those things. Check the p-value. I already explained you in the SRI Matucan, the adjuctor, the square. Those slides are just for stabilizing what I have tell you already. After normally when you make a fit you have to check what we call the almost-cadasticity. Normally the hypothesis of the linear fit is that our data has the same variance in all the domains. So usually I really make usually because most of people don't do it but we should check the residue and check that the residue is normally distributed. This is our usually statistician analyzed linear fit. If the residue is not normally distributed it puts some question on your fit and usually you should try to find transformation of your data so that you transform your data in a data where the error is almost-cadastic. That means that almost-cadasticity is the variance. It's also almost-cadasticity. Meaning that the variance is always in your domain the same. If it's not the case it's heteroscedasticity. There are also something you can and you have to check especially when your design is not regular, it's not a good design that you do not have one experiment which is influencing too much. What you would like is to have an estimate that was influenced quite equally by all your data. If you have and we call that leverage, if you would like to estimate a linear fit and you have a lot of data in one point and just one in another position, these new points have a big leverage because the value of these new points is making all the slope of your line. It's something you want to avoid. It's evident when you have one dimension, when you have more than one dimension it could be quite complicated to check just graphically. It's

notes

summary

25m 19s

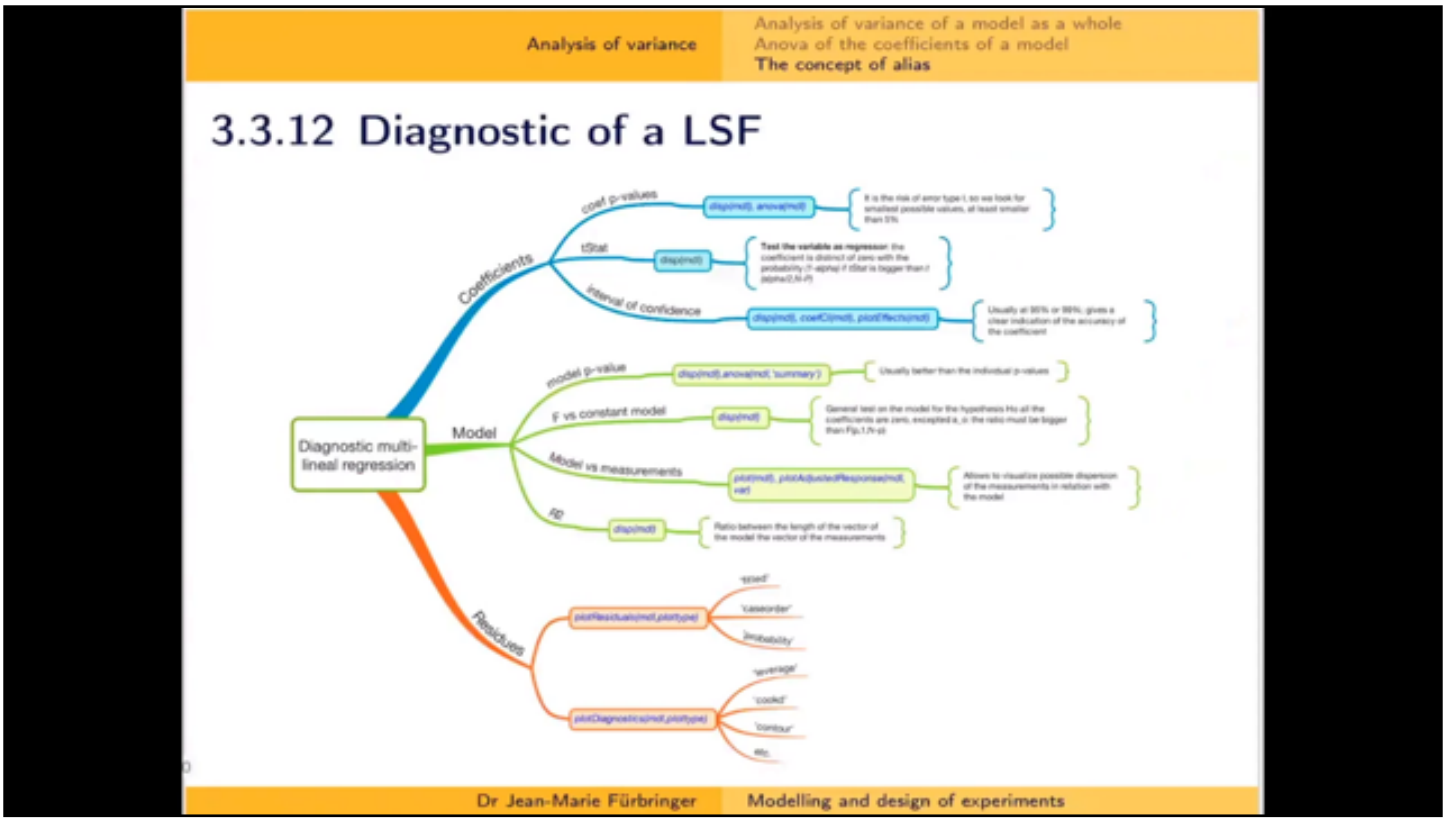
3. Homoscedasticity and Leverage & Influence

- **Homoscedasticity (Constant Variance) :**
  - Residuals should have constant variance across the range of predictors.
  - Breusch-Pagan test can formally check for heteroscedasticity.
- **Leverage and Influence :**
  - Use leverage statistics to detect points with large influence on the model.
  - Cook's distance can help identify outliers that may disproportionately affect the model.

what we call the check of the leverage. For that we have something that we call

notes

summary

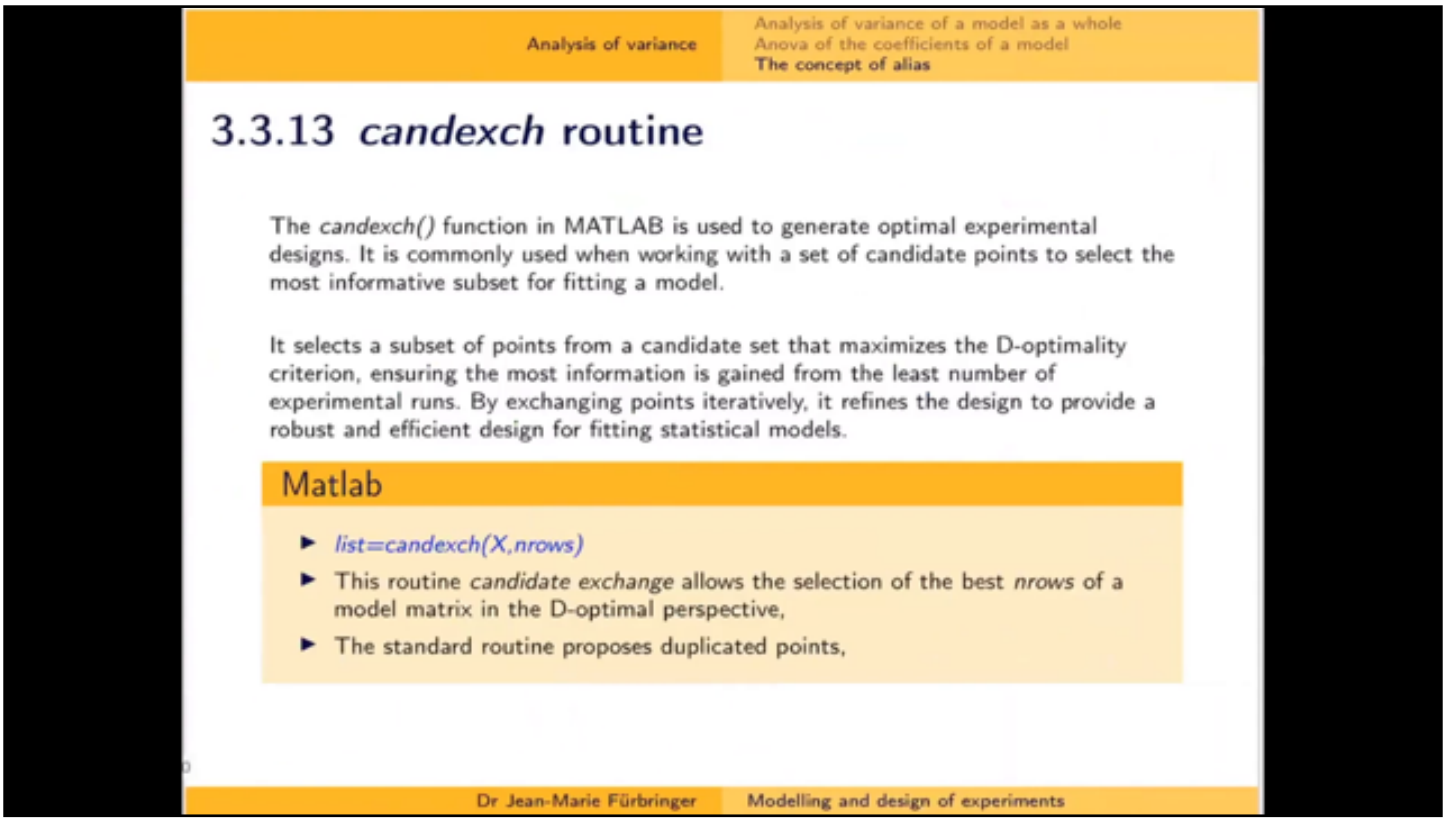


the cook distance of the points and you can check the cook distance. I don't want to spend too much time. For me it's not design of experiments, it's your cue statistic. But if you have tricky data, be careful with those things and spend a little bit more time about those concepts. It's just for avoiding that you have one or two points determining quite all the slope in your model which would be quite dangerous. It's okay, you can make a fit but after

notes

summary

28m 1s

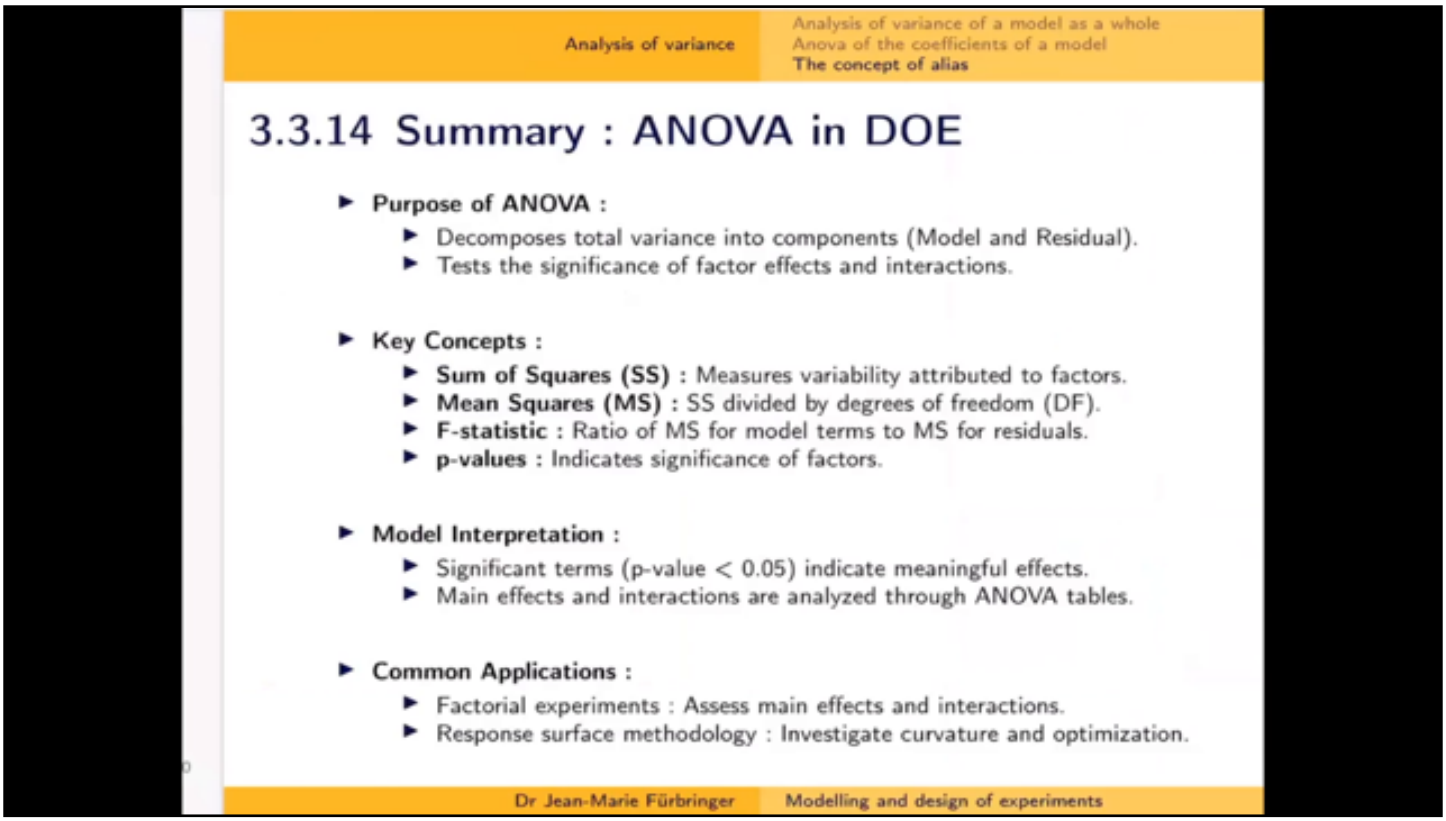


check the quality of your foot. Here you have a sort of diagnosis on the linear fit. It's just repeating the different things I have already said but it's an image you can keep on your side in your bed. Every night you can check what is a good fit and what you have to check when you make least square feet. Yeah, probably in most of the case your linear feet are good. Be careful with the tricky situation. Be aware of what you have to check. I'm not doing it so I'm not obliging to do it also. I don't check all the time everything but be careful what could be the tricky aspect of the linear feet and don't fall because referees in your papers could be quite harsh if you present linear feet with some problem. When you are refereeing it could be also nice to check one of

notes

summary

28m 39s



those things just to be sure that you are not in a very bad situation. There are a final routine interesting could have been useful for the last exercise but now you know that a routine is doing what you have tried to do for the last exercise. Candex or candidate exchange. It's a very interesting routine. You can fit this routine with all your possible experiments and say okay here you have my 20 experiments. I just need 10. Select me the 10 best experiments. So it's very interesting routine. And here you have then a summary of my chapter and so after the holiday we will start looking at the designs a very interesting design. So in this chapter we have talked a lot about ANOVA. It was the objects the principal objects what we check after we've made measurement and how do we check a linear fit. So we need to decompose the models in different layers that interest me for qualifying that and we need these layers are orthogonal between them for saying something. And we have talked about the different tests. The last point is more related to the future. So we will see what is the implication for on this was the what we call the factorial experiment. We will spend one week perhaps even two weeks looking at the different factorial experiments and the family of the factorial experiment when we make measurements at the corner of your domain and after we will see a few standard design with what we call response surface. We will travel for making the best surface in multi-dimension and I will show you a few designs that are very very interesting for doing that. For investigating the curvature of your model and also to see where you have your minimum and your maximum. Okay this was my last slide for today.

notes

summary

29m 42s