MITOCW | Lec 15 | MIT 2.830J Control of Manufacturing Processes, S08

DUANE BONING: OK, so last time we continued with our discussion of design of experiments and especially looking at fractional factorial designs, some of the aliasing patterns that come up, and how that interplays with model construction, in particular, what terms of a model you can include, what you can't include, as well as a few ideas on different kinds of patterns, things like the central composite pattern as well as fractional or full factorial. What I want to do today is pick up a little bit more on response surface modeling, or RSM. We've already touched on some of these, but there's a couple of things I've alluded to, but we haven't really shown you, things like how one gets confidence intervals on the estimates of coefficients in the model.

Just like when we were doing some estimation of statistical distributions, we would say we want more than just an estimate of the mean or an estimate of the variance of a process. We would like to know what range we might say 90% of the time with 90% confidence, we think the true mean or true variance lies. Similarly, when we're fitting models and model coefficients, we'd like some notion of what range we think the true model coefficients likely lie based on the data that we have.

So I want to go over that a little bit. And then we'll start talking about using these models for process optimization, so combining a little bit of the response surface methodology with design of experiments both in sequential fashion and in iterative fashion, where one might adapt the model on the fly or based on the additional experiments in order to drive the process or try to seek out and find an optimum in the process. So that's the plan.

I've noted here a reading assignment. You can read all of chapter 8. It's actually interesting, but what I'm mostly focused on are the first three sections in May and Spanos, which talks about process modeling. So it's covering a lot of the material here on response surface models, model fitting, a little bit of regression, and then also using these things for optimization. So a couple of chapters that have a little bit more advanced material on principal component analysis, which we may come back to a little bit later. OK, so that's the plan.

Here's a list of some of the fundamentals of regression. When we were talking about fractional factorial and factorial design, especially those formed out of contrast, that simplified method using differences in different collections of the data, we found that those were very useful, quick ways to be able to estimate model effects, to fill those into ANOVA tables and decide if those effects are significant, and then also the relationship of those for model coefficient estimation. I want to talk a little bit about the alternative perspective, which is regression as a way for fitting those coefficients. And we've already done some of that.

What I'm going to illustrate here is our basic assumption and what falls out of using minimization of least square or squared error estimates in order to fit the coefficients or estimate the coefficients in a model. And I want to talk a little bit more about estimation. We've already touched on estimation using the normal equations. But especially I want to talk about the variance again, in these coefficients, things like the confidence intervals for fitting of coefficients.

I'm going to do this here mostly in the context of a simplified perspective, a one parameter model. I just have one input and one output. And we'll do the simplest model. We'll build it up to a simple linear model, but all of these ideas also carry through for polynomial regression when I've got multiple inputs. But I think it's a little bit easier to see and discuss in the context of a simplified-- a simplified model.

And we also talked last time a bit-- the last couple of times about lack of fit. And I have a little example that carries us through the development of a model looking for lack of fit or seeing lack of fit and extending the model. So it's got a small example embedded in here. In fact, that small example might look familiar to those of you that saw or took 2853. It's actually the same model that I described in a very condensed lecture there on regression.

It's also important, I think, for us to get a little bit of terminology. You've probably run into measures of moral goodness, an overall summary measure of R-squared that is an attempt to capture how good the model is in describing what's going on with your data. So once one is done, the ANOVA analysis it's actually quite easy to calculate both the goodness of fit R-squared and the adjusted R-squared as shown here because they both depend on-- both of these R-squared measures and the ANOVA look at the amount of variation in your data and the amount of variation expressed in your model and use those to summarize how good the model is.

So the first measure, this R-squared, is basically just looking and saying if I were to simply model my output as the mean, how much better does a model that has more than the mean in it do in explaining the data? So essentially what we do is look at the sum of squared deviations around the mean. So this is total summit squared deviations around the mean.

And then we say OK, how much sum of squared deviations based on the model, so the amount explained in the model, compared to the total deviations around the mean? What fraction of those is captured in the model? So in other words, if there's really nothing going on except a flat dependency, that is, there is no slope with x.

As I vary x, nothing changes in the model, then this simplified notion of R-squared is basically saying there is no dependency on x. And therefore, I'm going to explain with the model essentially nothing. Now it's funny because we are ignoring the fact that you might also be fitting the mean value. But the notion captured in the R-squared is dependence on the input, dependence on x.

Now the big gotcha with this simple measure is I can always add model coefficients and fit more of my data, or at least I can do that ignoring replication in the model. For Example, we saw with a, say a two input, one output model and a full factorial, If I just have those four corner points, I can fit up to a second order model with the interaction terms. If I have four data points, I could fit the mean, first order, first order, and interaction with exactly four coefficients.

And in that case, what would the R-squared be if I put my data with all four coefficients? 1. I would fit the data perfectly. Again, this is without replication. And I would fit the data perfectly. And therefore I'd have an R-squared of 1.

Now is that really a perfect model? Well, kind of, but what you've done is you've used all of the degrees of freedom in the data to actually fit or use them to fit the model. We also don't have any notion of replication, which isn't really completely captured.

So one way of penalizing ourselves for the use of these additional model terms is to essentially have a different perspective referred to as the adjusted R-squared, which essentially looks at the residual data. Rather than the deviations captured by the model, it's looking at OK, what deviations are not captured by the model? What residual data would I have, which also has a side effect of essentially penalizing us for the use of additional model coefficients because we use up degrees of freedom in the model when we add model coefficients.

So very often people talk about the adjusted R square as this fair comparison between models, especially between models where I may have a simplified model with fewer coefficients and a more complicated model with more coefficients. And essentially what we do is form it as the ratio of the mean square error of the residuals over the total mean square variance, if you will, captured by deviations around the mean and then subtract that all from 1. And the way I like to think about it is essentially, I start with the perfect model. And then any residual error, which could include both replication error and lack of fit error, whatever percentage error that I don't capture in the data-- the sum of squared deviations divided by the degrees of freedom, that's my mean square error estimate or my estimate of the true total variance around the mean.

Whatever fraction of that that is in the residual, that's what I'm not modeling. So essentially what we're doing is simply looking at what's not expressed in the model. And the model can never capture pure replication error, so it's got that variance, but it might also have lack of it in it.

So most statistical packages will report out both of these numbers. You can also calculate them. But it's generally I like the R-squared adjusted as a better measure. In part because it feels to me a little bit more conceptual and comprehensive, in terms of telling me what's not captured in the model, how much pure variation going on in the data is not in the model.

However, you have to be really careful interpreting what that R-squared is telling you. It's not necessarily telling you that your model is good or bad. You might have a perfect model given variant noise factors in the model. So for example, if underlying everything, I've got a true systematic dependency, but I also have pure replication variance, that's going to limit how good your R-squared can possibly be even if your model were perfect in terms of capturing the systematic dependency. I think there was a question lurking there in Singapore.

AUDIENCE: Yes, so for R-squared and just R-squared the closer those values are to 1, the better of the model is?

DUANE BONING: Yes, definitely.

AUDIENCE: OK.

DUANE BONING: Definitely 1 is better. But you have to be a little careful in interpreting because even--

AUDIENCE: What you just-- no but what, Professor, you just said is the R-squared increase both the error of the model, also the error of the noise. So you can't really differentiate between these two.

DUANE BONING: That's right, that's right. And that's where a lack of fit analysis-- and we'll go in and do one of those as well --is also still important for being able to try to differentiate between those two sources of imperfection in the model. Yeah?

AUDIENCE: Also you mentioned the second R-squared also being [INAUDIBLE].

DUANE BONING: Right.

AUDIENCE: Your main concern is fit and having more coefficients is cheap, would you prefer R-squared or adjusted R-squared?

DUANE BONING: So the question is what would I prefer if the number of-- if fitting additional coefficients is cheap.

AUDIENCE: And fit is more important

DUANE BONING: And fit is more important. I think I would still essentially think of R-squared as somewhat of a more representative description of the trade off between adding coefficients, improving my fit. But also my R-squared doesn't get as much batter.

And in fact, if I start overfitting, it will tend to degrade slightly my R-squared. However, what I think a better mechanism for actually making the decision about whether to include coefficients or not is an analysis of variance and looking at the significance of those model coefficients, both the significance and the magnitude. So I would tend to do more the regression analysis together with the ANOVA. And the R-squared is a nice aggregate measure, but it's not the thing that drives my decision-making so much, so I hope that helps. So we'll see some examples of some R-squared that come out of some analysis.

Now we said that regression is at least as almost-- most commonly used is driven by minimization of a least-- or minimization of a squared error measure. And this is just trying to illustrate what I'm talking about here with where the residuals, the differences between my model and my data, may come from in the simple 1D case. We've already talked a bit about this, but I'm using a very, very simple model here, which has only one term.

It doesn't even have a constant offset. It's simply got a linear, direct linear dependence of the output on the input. And I'm saying that the true model does have some noise in it, which is normally distributed. And I'm fitting that or estimating that with some coefficient, a little b.

And so this is my fit through my data minimizing the squared deviations, or I'd like to minimize the squared deviations. And again, we're saying that any differences between the model prediction, essentially the y hat sub i minus the y sub i for that data point, that's a residual. That's an error. And it can come from two factors again, either lack of fit in the model or because of the underlying noise in the data.

Now last time, or maybe even 2 times ago, we talked about the use of regression numerically, if you will or algebraically, to estimate this beta with the best be based on a minimization of the sum of squared errors. So we take each one of those residuals, square it, and then we sum that over all of our data. And it turns out what we're trying to do is find the beta hat, the b that estimates the beta hat that minimizes that sum of squared deviations.

And what's nice with linear models is there's an algebraic way to find actually what b does that minimization for us. But I also want to just remind you that lurking inside of that minimization is an estimate of the total sum of squared residuals, SSR, what's lurking back there in that R and R-squared adjusted. And then if I divide that out again by the degrees of freedom, mu sub R, then I've got also my estimate of variance in the underlying model assuming no lack of fit.

So we said with least squares estimation, I can form the set of linear equations. And assuming that the residuals are all normal or orthogonal to each other, then the sum of the product of our residual and the input should sum to 0. And when you carry through the algebra for that, out plops the formula for the slope coefficient given our data, simply the sum of the product of my x sub i times y sub i over the sum of my x sub i squared, it's funky.

And as I said, here's our estimate of the underlying variance. That's our best estimate, unbiased, best estimate of the process variance. And in this case, we're only fitting one model coefficient. So I've got my total number set of data and then I've just got n minus 1, since I've only got one model coefficient.

Now the interesting thing that I've alluded to in a previous lecture but haven't shown you is I want more than just the best estimate of b. I'd like to have a confidence interval on b. Given the spread in the data and an underlying normal noise model or noise assumption, what do I think the range, say a 95% confidence interval, might be on my estimation of b? And we can do that very simply by taking the formula for b and simply doing our variance of b calculation on that formula.

It's just variance math. And that's what's broken out here in the y. If I expand out the b summation into a some of those individual terms, I can then apply my normal variance math here. And what I've got for the variance of that some-- just thinking of each of these elements has some constant, then that the variance of that sum of terms is the variance of the constant squared-- or excuse me, the value of the constant squared times the variance of each of those underlying variables. And when you go and do that, what you've got is another formula down here for the variance in that coefficient b based on the data that you've got.

So once I've got that up here, I've got my estimate for the variance. Now we've got an estimate of what 1 standard deviation would be in the variance. And then you can express that based on whatever confidence interval you want. So I might write that typically as b plus or minus 1 standard error, 1 standard deviation in b.

1 standard deviation-- I can't remember, what that correspond to, the typical? Got about a 90% confidence interval? Plus or minus 1 standard deviation? 67%, thank you.

The one I always remember is two standard errors. That's about 95% confidence. So if you wanted to 95% confidence interval, now you know how to formulate that. It might be 1.96 or whatever it is.

So there you have nicely falling out of the basic mathematical formulation for minimizing the sum of squares both the best estimate for your slope and a confidence interval to the slope. By the way, if you're based that on a relatively small number of data points, you should probably use a t distribution rather than a normal distribution. So it might change my 1.964 for a 95% confidence interval, as we're used to.

So this also lets us now go back and do-- think again about another perspective on analysis of variance. In fact, you guys played with this a little bit or saw this in a slightly different form on the quiz. There's two ways of thinking about the significance whether some slope coefficient or model coefficient should be included in the model. The basic hypothesis is are we saying do I have enough evidence to suggest that that slope term is non-zero? If it might be 0 to some degree of confidence, then I shouldn't include it.

So one way of doing it is the ANOVA with the ratio of variances in the F test. The other way is basically looking at the confidence interval for beta, say the 95% confidence interval, and if that intersects 0, that says that more than 5% of the time based on just random variation in the data, I might have a 0 coefficient there, in which case I cannot say that it is significantly different than 0. So you can make that determination about whether you should include the model coefficient based on your confidence interval for each individual term as well. So that's just alluding back to what we already know but just trying to make sure you see the connection or alternative ways of looking at it either in the ANOVA table, or if you want to look at individual coefficient terms, the confidence intervals on those individual coefficients.

OK, let's do an example. Here's a very simple set of data. We've got some input, some x value, call that "age". And some y values. Call that "income".

And if I just plot the data, let me get the data up here-- actually, what I've done here is used JUMP. I don't know how many of you have played with JUMP, but I love JUMP because it's nice and interactive. It does a lot of regression analysis, lets me explore the data fairly interactively, I like it a lot better than Excel for doing some of these analysis.

I think in an earlier problem set, we did give you a pointer to where you could run that on Athena and so on. And what this is doing is basically looking and doing my analysis of variance for a very simple linear model without a constant term. So I've just got one model coefficient, looks at the sum of squares, the mean square, looks at the residual with the remaining data point forms and F.

That F ratio is huge. It's 1,000, and the probability of observing that large of an F is minuscule. So I have great confidence that in fact there is a slope.

And if I look down here at my income leverage residual versus the age parameter, I can see this is basically just y sub i x sub i. I see a definite trend there. Now what this nice plot has done is the solid line is my best fit, but it is also plotted for us with the dashed line the confidence on the output. I think it's a 95% confidence interval on the output as well. Now I told you how to get an estimate on the confidence interval for our b term. How do we get a confidence interval on the output term?

Well, what we're going to need to do is also do the variance calculations on our y formula and see how uncertainty in our data also propagates through to uncertainty in our output. But before we do that, we can also see here in the JUMP output things like the parameter estimates for our age dependence. So here's our best guess for the age dependence is a simple 0.5 estimate.

And it is also showing us things like the standard error in these typical ANOVA tables, which we've ignored in the past if you've been looking at these. But that can also be used them directly, as we talked about, to give me a confidence interval, depending on what level of error whatever level of alpha I want to be able to estimate those things. And it's also looking at an individual t ratio for each of the coefficients.

I've only got one here, but it's basically doing a one by one assessment of each of my model coefficients to see if it's significant. And in fact, it's significant since it's exactly ends up being the same probability not really shown here. Essentially, the t test and the F test are identical in this simple example.

AUDIENCE: [INAUDIBLE] think of some subset of data, wouldn't it make sense to have [INAUDIBLE] part of the data then use some for testing like the model and seeing if it actually has a prediction because if you use that entire data set then essentially--

DUANE BONING: That's an interesting point. So what you're saying is how about the idea if you have a fair amount of data of holding out some of the data, fitting the data some portion of it, and the held back data to sort of test the model. And I think, especially when you do nonlinear models-- and I don't mean just polynomial, but I mean some other nonlinear dependence --that cross validation is extremely common and very useful.

Here, you could do that. And essentially what I think that's doing is allowing you to do a lack of fit versus noise estimate. In other words, what you're doing, I think conceptually, there is saying here's what my model would have predicted.

Here's my data point. There's a residual that I'm going to attribute maybe-- again, it's to a mix of random noise underlying but also model lack of fidelity. I think it's more common to go ahead and use all of your data because then you've got your aggregate measures and can run all of your tests with the highest resolution possible. But I suspect there's actually a relationship that's very close in there.

I think it's a little better to use all of the data because the more data you have, the better your estimates of underlying process variance are so you can better differentiate lack of fit from noise. But I haven't thought about that very much, especially of the simple linear cases. It's an interesting approach.

So I want to come back to this lack of fit versus the pure error because we talked about often being able to do multiple runs at the same x values. In this data here that I've shown you, we actually have a difficulty in distinguishing between model lack of fit and underlying variance. I had to basically make an assumption that my underlying model was truly linear.

And then I'm basically assuming, if I go back even further here-- where did my data go? --I'm basically assuming a y sub i is equal to beat sub-- beta x sub i plus epsilon sub i model. Why not-- I have really nothing except ideas of parsimony, simple models in general and perhaps prior knowledge of the physics of the process to really say this is the form of the model.

If you look at my data, why couldn't my model be that? It may well be. It might have a very complicated structure. That might be true.

The problem is I don't have-- in this random data, I don't have any replicates to be able to give me an independent notion of underlying repeated variance noise from model form. And so that goes back to what we said is if we have multiple runs at the same x values, especially if we design an experiment so that we do that, and we aren't using this sort of happenstance data, then we can decompose the total residual error into that lack of fit and pure replicate error and start to be able to distinguish between model structure and and pure replication error. And so we talked previously about being able to form the F test, the of variance explained by deviations from model prediction in the replicate data over total error and then seeing how likely it would be to observe that ratio and use the F test in the ANOVA test for that. And we'll come back to that a little bit in an example.

This is a quick one. I showed you an example here where the previous example was a pure linear term without even a constant offset. We can also do models that have both a slope term and a constant term. And this is simply formulated here as a means centered model.

If I were to take my data in and say when x was added mean, this term would be 0. So this is not really an intercept. This is saying my a coefficient is when x is added to mean.

I could similarly formulate it so that the coefficient would be when x was 0. The point being that the same approach for estimating both a linear term and a constant offset term can apply. And the same notion of not only getting estimates but also getting confidence intervals based on variances in those coefficients applies. So we can also use this to get confidence intervals, not only on the slope term but also on the variance term-- I mean the offset term.

Now we can also, what's nice is, do the same math now and look at a variance in our prediction of the output. I already alluded to that with these confidence intervals on that plot of y versus x in that one set of data. And if I basically am saying, OK, this is my best estimate-- this was my-- this is equal to the a coefficient --this is my best estimate of the underlying linear model with an offset term, and I just do my variance math on this, I've got a variance of some of these terms. And if you carry through that math, this is just a constant at each x sub i. Since x bar is a constant, x sub i is a constant.

So in the variance math, when I look at the variance of this term, it's the variance of this times the variance of this. This is a constant term, so I've got that constant squared out in front of the variance of my b. We already calculated what the variance of the b a and the variance of the B term were.

I can plug those in and get an overall estimate of the variance of each of my y sub i terms in my model. And based on-- once I've got that for the single standard error, my single standard deviation, I can use the t or the normal to get a confidence interval on the output. So it's the same thing we did on the coefficients. I can also do it to tell me what kind of spread, what confidence do I have in where the true output should lie when I'm predicting for any x value, where I think the actual true output y would lie.

Now there's an interesting aspect to this, which is if I look for any given x sub i input particular x input value, notice OK, that's right here. I plug-in for my particular i of interest. Notice that the denominator here was a sum over all of my data.

So that ends up being just a constant. It doesn't change. But depending on what x I'm looking at, where I am on the x, the size of this changes.

So for example, if I look at my mean, if I look where my x sub i is equal to x bar, that numerator term goes to 0. And essentially what I've got in that case is at the mean of my data, my estimation is basically-- my variance in my output estimate is basically just related to the random noise in the data. But then as I get further and further from the mean, my confidence interval in my output spreads.

So what you will often see on data-- this was x data and this is my y --is near the center of your data, you've got the narrowest confidence intervals. And as I get further and further away, if I were to use the dash for a 95% confidence on the output, the further away that I get in x from my x bar, the wider my prediction error becomes. Even though I'm still may be interpolating over the data I've got, my variance does spread as I get further and further away, just an interesting fact.

All right, we're almost ready to do a polynomial example. I just want to point out we talked about this previously. We can also do not only a constant term but also a linear term.

We can do terms that include this square polynomial, for example, include curvature in the x squared. One important fact is this is still linear data in the coefficients. And what this means is the least squares approach-- least squares minimization, still applies.

So you can still do least squares minimization to estimate your beta coefficients. And essentially what you do mechanically, say in something like Excel, is create that additional fake column of data, just taking your x. You can almost think of this as equating that with an x2, think of this as an x1, and building your data column, taking each of your x coefficients, squaring it, and that becomes a new x sub 2 input. And then all you're doing is just a linear fit now in these multiple coefficients. So it looks exactly the same like we did for multiple inputs, even if we have additional higher order terms in the x squared.

So let's look at a simple example here. Pull these threads together, look at confidence, but also look at it in the case when I've got some replicate data so we can get a little experience with this lack of fit idea. And so in this case, we've got importantly here cases where I've replicated my x values.

So I've got two runs with 20 grams of some kind of growth supplement. And so I've got two different output values at that point. And I've got another point where I've got three replicates, triply replicated set of data. And what I'd like to do is try to fit a model and hear what we've got in the picture is an inkling or a foreshadowing of some of the kinds of models we might consider and some of the issues we might consider.

If we look-- I think you can see it here --the basic data here in black, these are the data points. So this is just my output. There's my triply replicated data. There Is my x data.

First off, I could try to fit that with a mean. That's just the red line. That's the pure just mean of my data.

The green line here is a first order fit to just a slope coefficient and the mean, so two model terms. And you can see already that's not going to be a very good model. And what we've got is enough data here with the replicates to perhaps be able to detect that using our machinery of ANOVA, and then perhaps then build that into a second order model that we can already get a sense is going to be a quadratic model that fits the data lot that a lot better.

Now, If I were to just try it-- let's say I didn't already-- first off you should always plot your actual data so you have a feel for what kind of a model is going to be needed. So if you were to actually plot that data, you would already you probably needed a quadratic model. So you might go ahead and up front, include that term. But let's say we had not done that, we'd just tried to fit it with a very simple model, a simple linear model.

And if we go through and do the ANOVA, now because we do have repeated residual, I can split my overall residual sum of squared deviations into a lack of fit term. That's a sum of squared deviations just from my replicated-- or my total deviation from my model from my replicated data. And I can formulate then a ratio of those two things. And what I've got is deviations from my model that are much larger.

So this is a deviation. It's not a good one. Actually right, there the deviation from the model is quite small.

If I were to look right here, for example, this is my deviation from the model. I don't have any replicate data there. Right here, I've got deviation from the linear model.

And then I've got pure replicate error. And you can start to see that the deviations from my best estimate prediction at the model is much, much larger. And that's what shows up in this ratio of the two variances.

If you do that and follow through with the F, that's highly unlikely-- that big of a ratio is highly unlikely to occur by chance given the noise spread. So if you actually go in and do the lack of fit analysis, it's already setting up big red flags. Here's my red flag saying, look out, look out. You've got a lot of evidence of a lack of fit.

What's interesting in this example is if I were to just look at the significance of the individual model terms, this pops out in fact that the mean is highly significant but the slope term is not. So this would say-- if I weren't looking at lack of fit and paying attention to that red flag, I might be tempted to say a very wrong thing. I might be tempted to say there is a significant estimate of the mean that's non-zero, but given the spread in my data, I cannot conclude that there is a linear dependence on my input.

My linear dependence on x could be 0. In other words, with that green line right here, that's a small slope that given the spread in my data is not justified to actually estimate as anything other than 0. Interesting, huh?

So you really need to look at both. I'd have to be very careful because the extra explanatory power of the linear term is very, very minimal here. So I might think OK, so I've really got no dependence at all, which what I really got his lack of fit. That making sense?

So what I might then do is say, OK, I am paying attention to that big red flag. I've got lack of fit. Maybe I better add a quadratic term, refit my data. So now if I look at the S for my model with the mean with a term for the linear coefficient and one for the quadratic, now what do I get?

And return to breaking apart my residual and now looking and seeing how much deviation is there due to lack of fit compared to underlying replicate variance. And now that ratio is very small. So now I don't have any longer any evidence of lack of fit, that's good.

And now I can return to deciding about whether individual terms are significant. And we don't see the full F test, it's an incomplete ANOVA. But what we would basically find here is the mean term is significant, the quadratic term is significant.

How about the linear term? It's still not significant. So in fact, we've got a a mean and a square term but no dependence on the linear term.

You will typically see that. In fact, these-- if these terms are truly orthogonal, if I add the terms, it should not change my estimates for the other terms. That's not quite true if you throw those missing terms into noise factors. But the basic point here is I've now actually captured that the dependence on x with this quadratic term.

So you can do exactly the same thing. This is the same data using Excel. And you get the same kind of a table here with an x term and x squared term.

And what's interesting here is you can also go in and look at estimates of the coefficients, the standard error, 95% confidence intervals. And I guess actually if you were to look at that 95% confidence interval for that x term, looks like it actually is likely to be non-zero. So I did get that right.

So actually you probably should include that term, even though the ratio is a little bit smaller. It is still significant. Now I also put this one up because it's also got estimates of your R-squared and adjusted R-squared. where it's giving you a nice feel.

R-squared of around 0.9, 0.95, you start to feel pretty good about-- pretty good about your model. So I don't know if you played around with Excel. So again, I encourage JUMP, but if you do need to use Excel, there is-- under the data analysis tool if you pull that down, you will also see the regression analysis.

And it will let you indicate what your output problems are and what your input columns are. And it does just the least squares regression, pops out your ANOVA table for you. In that case, you actually have to construct by hand your wide square or your x squared data if you want to polynomial fit.

And that's what I've just illustrated here. You can't simply, unfortunately, at least in the version of Excel I have, say I want to try a polynomial model up to some order and have it just know to do that on the polynomial input data. You actually have to create columns for each of the model coefficients that you want to estimate.

Here's the same polynomial regression using the JUMP package, again, with all of the lack of fit versus pure error, the x and x squared terms, t ratios, all of that, but basically the same analysis with the second order included. OK so with that, I'm going to-- about to move on to process optimization. But I'd like to take any questions on regression, confidence intervals, confidence intervals and input, confidence intervals and outputs. Is that all? It's starting to feel-- are you confident in your understanding of confidence intervals? Yeah, question?

AUDIENCE: Definitely don't know what do you do if your inputs that are correlated?

DUANE BONING: OK so the question was, what do you do if your inputs are correlated. So what is assumed in all of these fits is essentially you've got orthogonality. If we go back to the tables we were forming with full factorial and so on, we're assuming that each of your columns are orthogonal, which is to say we're assuming each of your coefficients in each of your different terms are uncorrelated or orthogonal.

If they are orthogonal, and you do a least squares regression-- or if they are not orthogonal, there they are correlated, what happens? Well, what happens is you've got to model coefficients both trying to explain some amount of the same data. And they fight against each other. And it's almost random how the effect that-- that true underlying effect gets apportioned between say a beta 1 and a beta to term.

In fact very, very tiny little perturbations, and you can get a different mix of beta 1 and beta 2. And it turns out you might still be OK in terms of predicting an output because at least your model has both of them in there. But it really screws up your ability to decide is that model term significant or not.

What you need to do is transform your data to get it into an orthogonal form to get rid of the correlation to basically create do model coefficients and new explanatory values to fake x values that don't have the correlation in them. And the classic tool for doing that is principal component analysis or some transformation of the data to a different basis than your original x1, x2, x3 coefficients. We might talk a little bit about multivariable things.

I think we did a little bit with multivariable statistical and T charts and so on, but essentially a principal components or some other kind of transformation is needed on the data in order to then have individual coefficients that are not duplicating each other. If you look, I think it's chapter section 8 point-- maybe 8.4. The next one after what I assigned as a reading, that talks about principal component analysis and how you do that and process modeling.

So you can read that section. It's actually very good, very interesting. Other questions, progression? Yeah?

AUDIENCE: If there is a big difference between R-squared and adjusted R-squared, what is that telling us? In this case, it's essentially [INAUDIBLE] 0.9 and 0.8, or 0.7 [INAUDIBLE].

DUANE BONING: Yes, so the question is what if you have big differences between R-squared and adjusted R-squared. I think it's essentially telling you that the influence of additional model coefficients is really important, both-- this very qualitative. But essentially, it's telling you there's more than going on than just the mean response.

So you're seeing a little bit of a mix of both-- the penalty of adding more model coefficients, but it's also telling you there's likely additional structure that you needed in order to use that. But that's pretty qualitative. I think basically it's signaling that there's more than just mean-- mean deviations going on. It sounded like there was a microphone question in Singapore?

AUDIENCE: Question on slide 50. You mentioned we should only see the mean which also focused on the lack of fit and the pure error. So why do you say that we only see the mean, we may say it's a good model. Can you explain that again?

DUANE BONING: Yeah, actually what I was saying in this example is that if I only looked at the mean, I might be hesitant to include any model terms beyond the mean. So I might not actually think it's a good model at all. So that part of your question, I'm not sure I quite understood or quite agreed with. But I do-- I guess maybe I'm just repeating myself, I think it is really critical to look for lack of fit because you need both perspectives.

You need to look not only at model coefficients in terms and whether they should be included in the model, but you also have to be alert am I missing terms. That's what the lack of fit enables you to do. This is basically saying the terms that are there, are they significant?

So in some, sense this one is basically just leading you to throw away coefficients and throw away model terms. And this number two, the lack of fit, is telling you, hey wait a second, there's stuff going on in the model that you're not explaining that's different than random noise, so maybe you should add model terms. And so you need both perspectives.

OK so I think we're ready to move on and look a little bit at process optimization. I want to touch on the most natural use of these sorts of models, which is we define an experimental design, we go gather the data, we build a model, and then we start playing with the model. I think of that is offline use of the model, using it to try to identify an optimal point. But it's not purely offline because I want to make the point that if you're predicting an optimum, you probably want to go back and run some confirming experiments and use those back with your physical process to check your model and maybe even iterate and improve your model.

So that's one natural approach. And the other is-- that should be online use. So another clever approach is actually build simplified models in a little part of the space, use that to tell me what direction to move in exploring my overall process space, and then dynamically build and improve my model.

In the case when my real goal is getting to an optimum, not having the perfect model covering all of my space but rather get to an optimum point. So I want to touch on both of these ideas, ways of using these sort of simplified response surface models. And part of the point here is one important use of these models really is trying to find an optimal process output or find the inputs that give me an optimal process output. And that optimal process output may have multiple characteristics about it that are important for us.

One is I want to be close to a target value. But the other is we may also want small sensitivity, small deviations in my output. And if we go back to our variation equation, that may mean I want small deviations around noise factors that I'm not controlling.

And I may also want relatively small sensitivity even to some of my input parameters because I'm going to fix them in my process. And I'm not dynamically or in a feedback loop changing them. So in some cases, I want this to also be small.

So we'll talk a little bit about ways to mix in these and other objectives. For right now, I'm going to mostly focus on say trying to meet some set of target mean values. But I can make the point you can generalize what I'm going to be talking about here by thinking of some objective function, or some cost function, or some goodness function that actually mixes in together multiple objectives.

So some of the objectives, you might have a cost function that penalizes for deviations from the target or maybe sum of squared deviations if I have multiple outputs from the target. It may also penalize me for larger x's because-- larger input because there's more cost associated with using more gas if I have a higher gas flow in some process. And then I can also include other things like terms that penalize for sensitivity, these delta y's, sensitivity to the output. And I can keep throwing additional things in.

So if I've got in general some complicated objective function, if I can formulate that and actually model either empirically or analytically that cost function as a function of my input or as a function utilizing the models that I already have, I can then formulate an optimization function or an optimization problem where I might be trying to minimize that cost or minimize that objective. Or maybe I'm trying to maximize it because I think of it as really a goodness function rather than a penalty function. But overall, I've got some complicated form for J as a function of my factors. Or my factors might be my actual input, but they may also be noise factors, other factors that I haven't explicitly modeled.

And we'll talk about robustness next week, or not next week, on Thursday. But right now, I just want to talk about adjusting or searching for good input factors to minimize or maximize some cost function with constraints. So in general, you can think about different approaches for this. If I've got a full expression for y as some function of x and maybe J is some function of y, I have overall got some overall function for my cost as a function of my inputs.

Then I can go in and try to minimize, really dj dx and find-- with some assumptions of monotonicity, I can find an overall minimum or at least a local minimum or maximum to that function. So that's if I've got a full expression. And we'll explore that a little bit.

Another approach is more of an incremental approach. Rather than having the full expression and leaping right to the optimum point based on a local minimum or local maximum, I may have to search for it. I may have to iteratively explore the space. And we'll talk a little bit about these with hill climbing or steepest ascent and descent kinds of problems. And I've already mentioned a little bit of this online versus offline.

So here's the simplest picture for one of these optimization problems. I've got my input x, and I've got my output y. And what I'm looking for is a maximum for my output y. And maybe here simply my cost function is simply J or J is equal to y, something like that.

So I'm not differentiating here too much between y and J. I'm just simply saying what I'm looking for is the overall maximum for this output. And one knows from basic geometry, basic algebra that the maximum will occur-- unless I hit some constraints or some boundary cases --will occur when I've got zero curvature in that function.

So how do I find it? Well, one approach is, again, this analytic approach. If I have a full expression, I can simply recognize that that minimum occurs where there is zero curvature, solve for the y such that that curvature is 0, and I directly get to the answer. But in order to do that, I need a full analytic model.

To do that, I needed perhaps relatively small or good accurate increments and x or assumptions on the model form. And especially if I have relatively sparse data points, if I had say just these data points, it's quite easy to miss the true optimum because of noise or imperfections in my model fit. So it can actually be a little bit tricky with small amounts of data to find that if I fit an overall analytic model to a very small number of data points.

An alternative is a little bit of an iterative or a search process where we might actually add data or explore or model, either explore experiments or explore a model in a smaller space in each case and sort of seek to find the optimum point. And here are a simple conceptual idea here is in some regions of my space, I may have very good model fits less so than with much less error than trying to fit this overall quadratic to a small number of data points. I may have relatively good model fit in smaller regions of the space.

Remember that confidence interval on the output? I said as we get further and further away from say the central moments of our data, my confidence interval on my output prediction gets wider and wider. If I shrink my space, I get better estimates of my model in a local space.

And so one approach here is to say, I'm going to look in a local space get a good estimate of what the slope is. Maybe it's a reduced order model that's only linear. So I'm not even trying to fit additional curvature.

And then use that to say my output y is increasing in this direction with x increasing. And use that to project forward a small amount and suggest a new x value to try. So it's projecting and additional steps to explore.

If I then do that and build an additional linear model-- whoa --build an additional linear model here, it might suggest another small step. And as my linear model starts to have a slope turn that shrinks, that's telling me I'm getting something closer to an optimum point or at least a local optimum point. And at that point that's signaling me that if I really want improved accuracy at that point in space, to really zero in on the maximum, I can do two things.

One is still constrained my search space. But also in this region, it's quite likely that my-- it's quite likely-- I don't want this. I don't know what that was.

Oh, wow, something funky happened. In this space, it's just like with that curvature model that I showed you earlier, the linear term is probably no longer very significant. I really need the quadratic term.

So I might fit locally a quadratic model just near the optimum which allows me in a restricted space to get an accurate model that really lets me zero in on the optimum point. So out here, a linear model might be good enough up in here. I may need a beta 0 plus a beta 2 x squared term, maybe still also with a linear term here as well. But I can basically build dynamically the model getting an accurate model near the optimum point.

Now, I showed you this in 1D, the point 1D, but you can also do this with two inputs, where I've got a 3D model if this is an x1, this is an x2, and this is a y. But you can essentially think the same thing. If I start out here in this space, locally it's linear. I can use that to suggest the next step to take using a simplified linear model in this region. And then as I hill climb up, as I get close to the optimum, then again now near the optimum, I need-- as my x1 and x2, I may need a quadratic model in those two coefficients. But I can extend the same idea to hill climbing not only in one input, but two inputs, three inputs, multiple inputs in order to get to an optimum point.

So essentially what we're doing here is, again, linear gradient modeling, it is useful often to include still an interaction term. But essentially we're doing exactly that same thing. And if my model itself is linear, an interesting thing happened.

Where is my overall optimum? If I'm trying to get to maximized y, where's my maximum y going to occur? It will always occur on a boundary when I hit a limit of my input and x's.

So an important thing that I haven't talked much about is also the notion of additional constraints. We may be driving to an interior point like in this model, but it's also possible that we may be driving to either a corner point or some other boundary point because of a constraint on my allowable ranges for my x inputs. There is another piece of terminology that's sometimes used for these kinds of searches, either steepest descent or steepest descent, whether you're climbing or looking for a local minima.

And the basic point is when I've got that simplified linear model perhaps with the linear interaction term as well, you can think about the local gradient with respect to x1 or the local gradient with respect to x2. And now when you make your step, what you often want to do is make the step in the overall steepest descent direction, changing both your x1 and x2 parameter at the same time. So this is simply showing when I move and hill climb, I may change x1 and x2 proportionally depending on the relative slope in those two coefficients. And it's relatively easy once I've got that model to decide what direction is the overall steepest descent.

Another point here is that with quadratic terms, you can have complicated functions where your minima may occur in the interior of the space or your maxima in the interior of the space. But you can also have hyperbolic or inverse polynomial kinds of relationships where, again, you may have local minima or maxima with respect to one variable depending on what you're doing with the other variable. Or you may also have places where you end up with a maxima again at your constraint points. So in your search, you've got to account for both.

So I can summarize what we've done here with a combined procedure for design of experiments and optimization in either the iterative fashion or at the end, I'll allude to evolutionary or incremental kind of version. So this is a summary of the last two or three lectures boiled down into a reminder-- a summary of the basic process or procedure for doing DOE and optimization. We said originally our goal here is to build a model, to do a design of experiments.

I do want to emphasize that depends on some knowledge of the process, a little bit of knowledge either experience based or in the physics of the process. Because you need that in order to do things decide what the important inputs are likely to be. Now there are things you can do with the DOE to confirm that or to expand your knowledge, like factor screening experiments.

We talked about fractional factorial with large numbers of coefficients where you're just trying to decide is there a main effect associated with that factor. But up front, defining the inputs is very important. We also need to define limits on the inputs. What space do we want to explore and build a model over in our design of experiments?

So overall, we're going to need to first build our-- decide on a DOE. We'd go and run our experiments. And then we're going to construct our response service model.

And if we're using it for the optimization, I also want to make the point that you need to think early on about what your overall optimization or penalty function is because that may strongly influence your DOE and maybe even your factor selection. So for example, if you believe that you're really going to need an optimization that folds in things like noise in addition to just trying to get to a target, that can have a profound effect on the DOE that you explore. And we'll talk about that on Thursday, where you might do additional small experiments at each point in the DOE in order to build a sensitivity model of that delta y as a function of some additional noise factors. So depending on what it is you're trying to achieve with your model, that can of course, I guess it's obvious, that can affect the structure of your model and the design of experiments that you want to do.

So we've already talked about a lot of this. Again in summary, your DOE includes decisions about what likely terms you think might be in there based on your knowledge of the physics. Is it going to be mostly linear? Might there be quadratic terms?

That can influence again the selection of the high-low center points. Do you need center points, do you need three levels for all factors, and so on. And you also need to think about things like the noise factors. We talked about these nuisance factors, if you will, or additional noise factors. So that you might randomize or block against those. If they're not going to be explicitly in the model, you don't want them aliasing with or confounding with the terms you actually had.

The response service modeling is actually a pretty easy piece, especially if you use things like the regression and the ANOVA approach. Again, you can use contrast, if you've got a highly structured design and experiment for very rapid estimation of those terms. But overall, the emphasis here is you're trying to determine if there's significant variation in your data, are individual terms significant, are you missing terms.

So that lack of fit is extremely important. And there's often a very interesting interplay with the regression modeling. In fact, an approach we haven't talked about much, but it's essentially inherent in what we've been talking about here is also referred to as-- I think it's-- not piece-wise, step-wise, step-wise regression.

And some of the interactive tools like JUMP actually explicitly support this, where one factor at a time, you look and say, I would like to add a term or drop a term based on cut off decision points, on significance, and so on. So you can build up an appropriate regression model by dropping or adding terms as needed. And we talked about this at a fairly high order or high level about the optimization procedure and again, just ideas of defining your penalty function and then searching for your optimization either piece-wise or analytically.

I'll come back to this in just a second. But I do want to emphasize that once you've come to some expected optimum point, you really should check that and confirm that often because you're building your estimate of your model based on relatively limited data, especially in the factorial models perhaps with only one interior point or center point based on mostly extreme old data. And especially if you've driven your optimum to some interior point using say the analytic model of the response service model rather than iteratively or incrementally, you're making a lot of big assumptions about the shape of the model right near your optimum, like it's convex right at that optimum point. So you really ought to go in and do a confirming experiment right at or right near your optimum in order to really test the model and consider model error right at that point. And that might actually drive you to improving the model or exploring slightly different space right near that optimum.

Now the one last thing I just want to allude to is an alternative approach here is often starting with some data point in a small space and building your model iteratively or adaptively. And next week, at the end of next week, we'll have a guest lecturer, Dan fry, who has actually studied one factor at a time incremental exploration and model building for the purpose of optimization a great deal. So he's going to lead us through an alternative approach of actually doing full factorial models but trying to find the optimum by not defining up front the whole DOE and running the whole thing, but rather just walking around your multifactor space in order to try to find the optimum point.

And that has some relationship to another approach that is also in May and Spanos in chapter 8.5 which I've just mentioned to you but not expect that you actually have to know a lot about, which is evolutionary optimization. Which would say build a local model use that again and a hill climbing fashion to suggest where you want to go for your next point. Maybe in fact you simply pick one of those corners. And then you build a do model around that. And it might suggest you move your process to another corner, in which case you build another model and so on, so that you can walk or evolutionarily arrive at an optimum point in your process, building local models along the way.

OK so next time, the one additional topic I want to mention in this space of optimization and process optimization and DOE is this notion of robustness. I'll allude to actually building models that include the variance in them and not just the overall output. So we'll come back to that on Thursday and enjoy.

In the meantime, I think you've got the problem that is due on Thursday. And it's going to let you explore a little bit more some of these DOE and response service model kinds of things. So we'll see you on Thursday.