Blocking and Confounding in 2^k factorial design: Analyzing the effect of grinding parameters on sintered reaction-bonded silicon nitride

Ritwik Dasika, Rishab Katteri, Ben Luckman, Saikarthik Tirumalareddy

Rutgers University

April 15, 2021

Abstract

In this experiment, we are going to analyze the ceramic strength of a certain type of high performance ceramic known as sintered reaction-bonded silicon nitride collected by the NIST Ceramics Division Material Science and Engineering Laboratory. Using the grinding parameters table speed (x_1), feed rate (x_2), wheel grit (x_3), direction (x_4), and batch (x_5), we can then reason and analyze how these predictor variables affect the strength (y) of the ceramic. We will utilize a different approach from NIST as we will first implement an unreplicated blocking design, then a confounded blocking design, and finally, compare the two resulting models using adjusted R-squared values, the Shapiro-Wilk test for normality, and a 10 fold cross validation. In the end, we discover that the resulting model that comes from confounding by the significant main effect term x_4 proves to be more useful than unreplicated blocking of 2 levels. Finally, we will compare the results of this experiment to that of the original experiment conducted by NIST's Lisa Gill of the Statistical Engineering Division.

Introduction

Ceramics allows many materials and articles to be made. It is defined to be any of the various hard, brittle, heat-resistant and corrosion-resistant materials made by shaping and then firing a nonmetallic mineral at a high temperature. While the earliest ceramics made by humans were pottery objects or figurines by sintering it in fire, many advancements have been made through modern industrialization and advanced ceramic engineering through events such as semiconductors and fracture mechanics. All in all, there are a lot of factors in designing and constructing ceramics such that the finished product holds its design and keeps its aesthetic.

What we aim to do in this experiment is to construct a 2^k factorial design which can best explain the relationship between the strength of the ceramic and the five factors as well as any interaction terms. A factorial design would best fit as we would want to fit our data as best as we can, which includes keeping or removing any n-order terms in our final model. Our dataset has these predictor variables that are of two levels, which furthers our reasoning in using a 2^k factorial design above any other model design tools. Fortunately, we are not dealing with many predictor variables which might have pushed us away from a full factorial design. The interaction terms are also prevalent enough such that we can construe how each of the predictor variables may be connected.

While this dataset demonstrated the usage of a full factorial design, we are building our model with a more detailed design approach. We will implement blocking in an

unreplicated design so that we can try and see how it affects model adequacy and overall fit. Another approach to improve the fit of the model would be to confound with respect to a certain predictor variable or interaction term, one which has a lot of influence on the model. The purpose of doing this would be to reduce the effect that this variable has on the response and the predictor variables. We would also try to use 10-fold cross validation to make sure our dataset will not overfit the in-sample data and adjust based on the noise if we include out-of-sample observations/data should we see to add any more. Following this, we would run a full residual analysis to completely confirm whether our model is absolutely adequate in judging the effect of grinding parameters when it is sintered in silicon nitride.

Towards the end, we don't expect there to be much of a difference between the chosen model from our experiment and the one used by Lisa Gill in the NIST experiment. Ultimately, we predict that the differences between the models will be negligible.

Materials and Methods

First, we read the data from the text file we obtained it from. For future reference, we will block the data according to our predictor variables table speed, feed rate, wheel grit, direction, and batch and add it as an ersatz predictor variable. Loading up our libraries of `conf.design`, `faraway`, `psych`, `FrF2`, `stats, olsrr`, and `caret`, we are now able to begin our designed experiments.

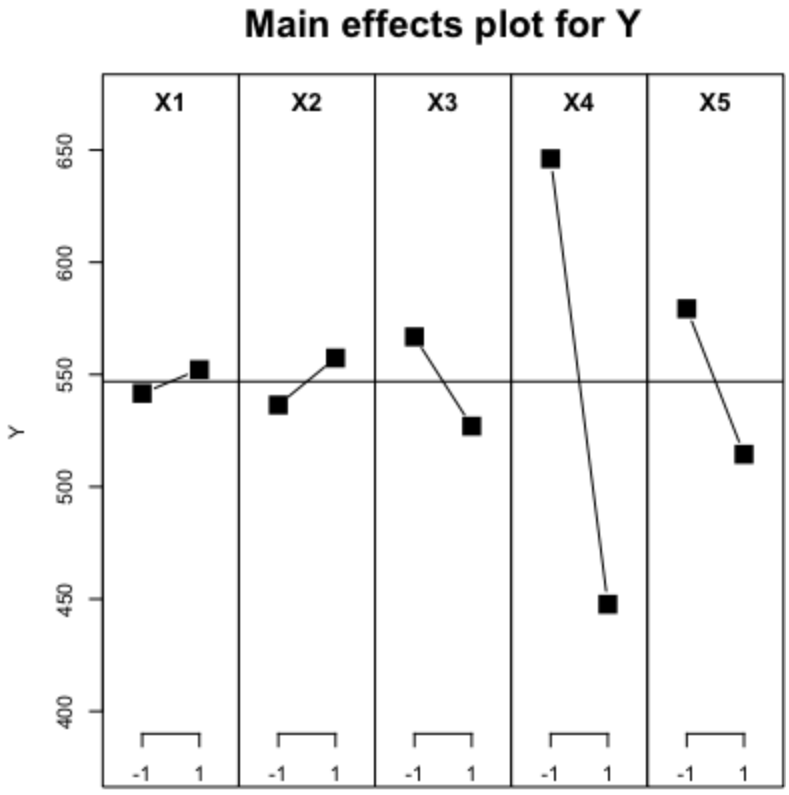
Starting from scratch, we want to see the full factorial model in its truest form. We run a model called `ceramic.model` and assign the ANOVA of all its predictor variables and all the predictor variables' interactions with correspondence to the response variable of strength.

```
> ceramic.model = aov(Y ~ X1 * X2 * X3 * X4 * X5, data = ceramic)
> summary(ceramic.model)
               Df Sum Sq Mean Sq
X1              1    894     894
X2              1   3497    3497
X3              1  12664   12664
X4              1 315133  315133
X5              1  33654   33654
X1:X2           1   4873    4873
X1:X3           1   1839    1839
X2:X3           1    307     307
X1:X4           1   1637    1637
X2:X4           1   1973    1973
X3:X4           1   3158    3158
X1:X5           1    465     465
X2:X5           1    199     199
X3:X5           1     29      29
X4:X5           1   1329    1329
X1:X2:X3        1    357     357
X1:X2:X4        1   5896    5896
X1:X3:X4        1      2       2
X2:X3:X4        1     44      44
X1:X2:X5        1    145     145
X1:X3:X5        1     30      30
X2:X3:X5        1     26      26
X1:X4:X5        1    545     545
X2:X4:X5        1    167     167
X3:X4:X5        1     32      32
X1:X2:X3:X4     1    354     354
X1:X2:X3:X5     1    356     356
X1:X2:X4:X5     1     79      79
X1:X3:X4:X5     1    233     233
X2:X3:X4:X5     1    269     269
X1:X2:X3:X4:X5  1    613     613
```

Because our design is a full factorial design with n=1 we cannot use lm() to make our model because our degrees of freedom for error are 0 meaning we would not have an estimate for error.
Thus, the only way to proceed would be to observe the effects of the predictor variables and their interactions from an ANOVA table.

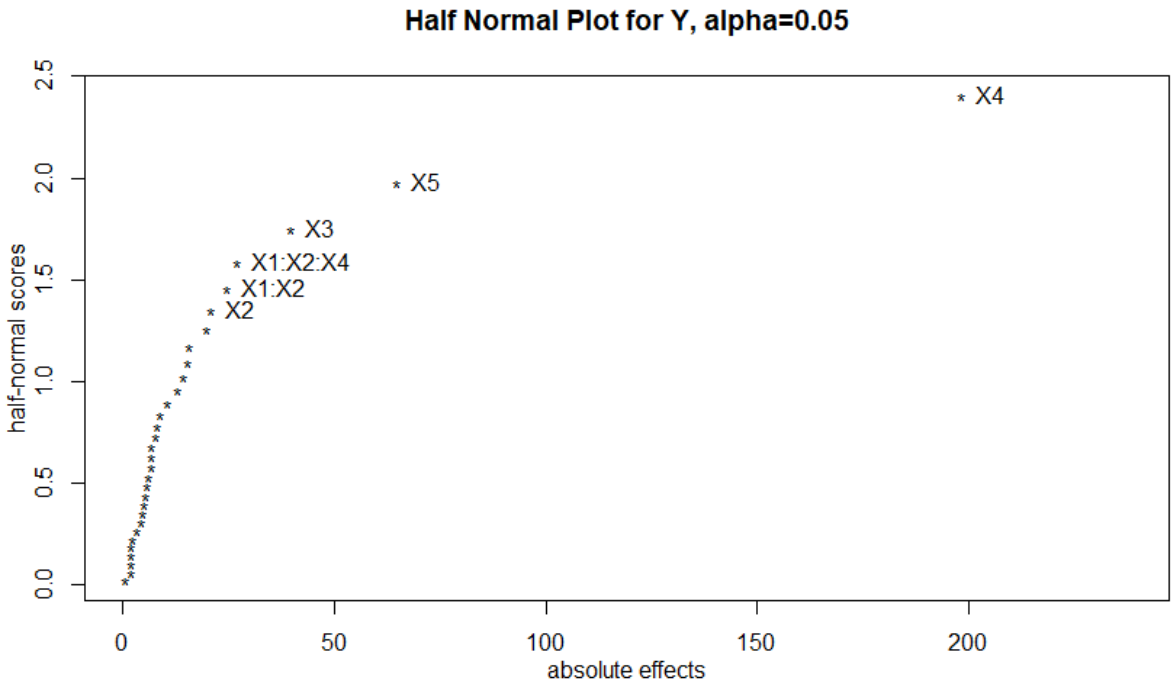
Just at a first glance from the ANOVA output, we can see that x_4 is the most significant main effect term in our initial model. It might be worth considering to confound by direction (x_4) as we move down in our analysis.

Let us use the `MEPlot()` function to print a plot of the main effects in this model:



We can see that clearly x_4 has the most influence on the ceramic strength, followed by x_5 while the other three factors have very limited influence.

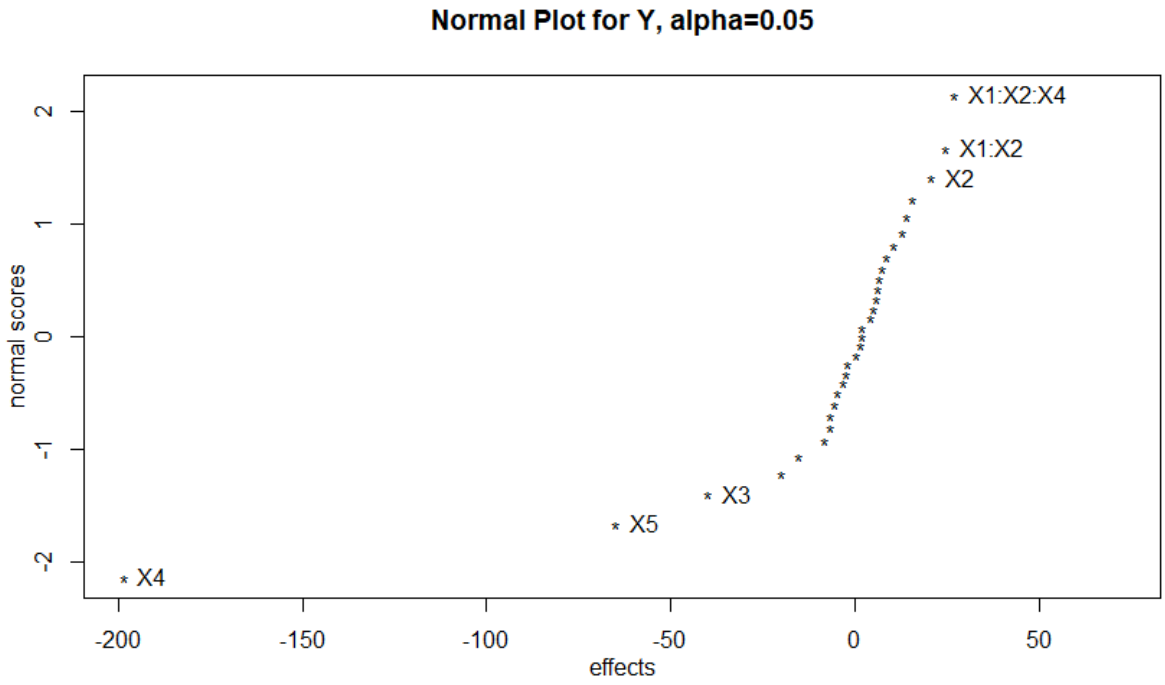
This is further substantiated as we use the `DanielPlot()` function to output a half-normal plot for Y, at an alpha level of 0.05, which shows us how important certain predictor variables or certain interactions are. Shown below is our half-normal plot. (Remember: the closer the points are to the origin, the less significant they can be)



Before analyzing the plot, the reason why we use the half-normal probability plot is to consider the important factors and interactions we have in our base model. Aside from the fact that it is typically used in 2-level factorial designs, We want to quantitatively show the estimated effect of a given main effect or an interaction and rank it according to other main effects and interactions through least squares estimation, or what we saw in our ANOVA table. We form these estimates such that it minimizes the sum of the squared differences between raw data and the fitted values. Having these estimates, we can then construct a list of the main effects and interactions ordered by how influential it is to the model. With our half-normal probability plot in hand, we can then assess which factors are important or which factors are unimportant according to our designated alpha of 0.05.

So, with that knowledge and motivation, we see that main effects x_4, x_5, x_3, and x_2 are critically important in our model down the line. The half-normal probability plot also shows that interactions x_1:x_2:x_4 and x_1:x_2 are also significant effects in this full factorial model. However, even though we do not see that x_1 is a statistically significant predictor of strength, we still must include it in our analysis, given that it holds some influence and significance in our model, justifying our inclusion of x_1.

What will confirm this is our normal plot using the DanielPlot function yet again. Shown below is our normal probability plot.



To capitulate, we will use a normal probability plot to determine the magnitude, direction, and the importance of the main effects and its interactions. Looking at this plot, we want to see effects that are the furthest from 0, as they are shown to be of statistical significance. At our designated alpha of 0.05, we can then assess the significance of the standardized effects and interactions.

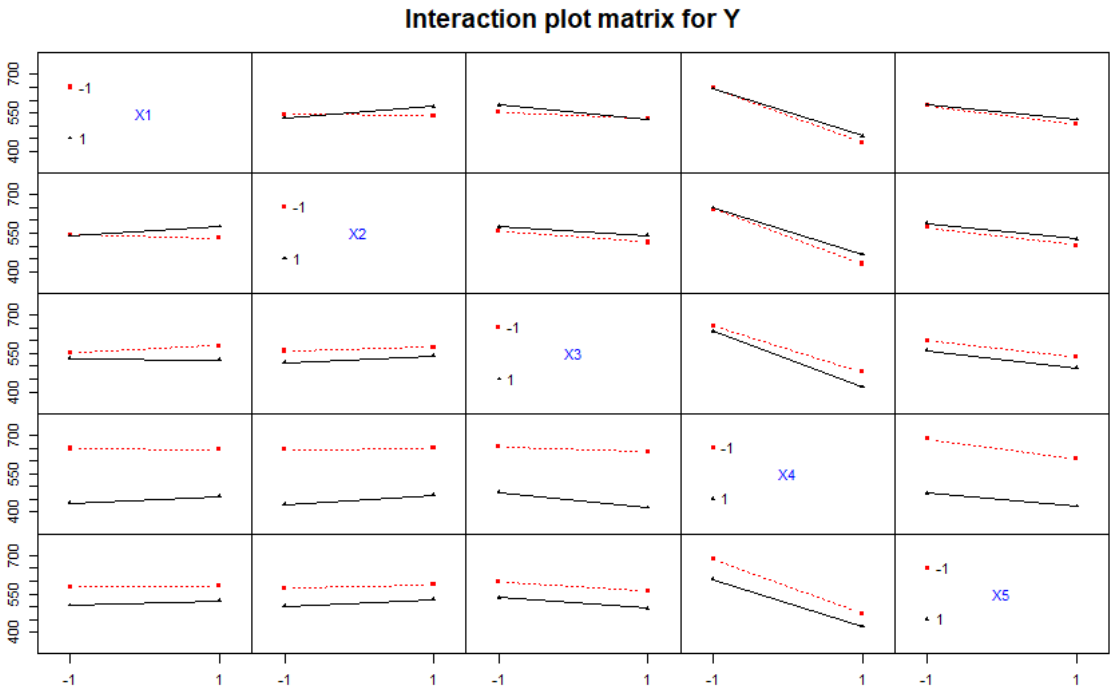
At a glance, we see what the half-normal probability plot told us, which was that main effects x_4, x_5, x_3, and x_2 are significant as well as interactions x_1:x_2:x_4 and x_1:x_2. We would still have x_1 as a crucial predictor variable to analyze our ceramic dataset, so including it is a necessity.

Now, we proceed to block our dataset with the main effects and interactions that we chose from our half-normal and normal probability plots. Since we only have values of 1 and -1, we would only need 2 blocks in this unreplicated design. After this, we will implement our confounding design.

Shown here is our next model `ceramic.2` with our chosen main effects and interactions added to our blocked data:

```
> # Blocking in unreplicated design with 2 blocks (1 and -1) NOT CONFOUNDING
> ceramic$Block = ceramic$X1*ceramic$X2*ceramic$X3*ceramic$X4*ceramic$X5
> ceramic.2 = lm(Y~X1+X2+X3+X5+X4+X1:X2+X1:X2:X4 + Block, data = ceramic)
> summary(ceramic.2)

Call:
lm.default(formula = Y ~ X1 + X2 + X3 + X5 + X4 + X1:X2 + X1:X2:X4 +
    Block, data = ceramic)

Residuals:
    Min      1Q  Median      3Q     Max
-41.655 -14.660  -3.702  11.430  57.598

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  546.896      4.295 127.334  < 2e-16 ***
X1             5.287      4.295   1.231 0.230811
X2            10.454      4.295   2.434 0.023109 *
X3           -19.893      4.295  -4.632 0.000117 ***
X5           -32.430      4.295  -7.551 1.14e-07 ***
X4           -99.237      4.295 -23.105  < 2e-16 ***
Block          4.377      4.295   1.019 0.318738
X1:X2         12.340      4.295   2.873 0.008593 **
X1:X2:X4      13.573      4.295   3.160 0.004374 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 24.3 on 23 degrees of freedom
Multiple R-squared:  0.9653,    Adjusted R-squared:  0.9532
F-statistic: 79.88 on 8 and 23 DF,  p-value: 6.136e-15
```

Off the bat, we see the tremendous adequacy of this model, having a p-value that is extremely close to 0. We see that our adjusted R-squared value is 0.9532, which is a very good indicator that the sample data is explained by the variation in the data. We also see that all of our predictor variables and interactions are statistically important sans x_1. It seems as if we would be satisfied with our result here and stop if we were not curious enough about improving model fit further.



Shown here is the interaction plot matrix using the `IAPlot()` function between all of the predictor variables. Nothing of what we see seems to be of statistical significance, as the plot produced pretty standard plots to follow. Again, it seems to be that the interaction terms in this model are all suitable for model usage.

However, there seems to be something more that we can do. Let us approach our dataset through confounding instead of blocking in our factorial design. From the half-normal and normal probability plots, we see that x_4 seems to be the most influential of the main effects and the interaction terms associated with the main effects. So, if we design our model based on reducing the influence of x_4, it might boast a better model adequacy and overall fit compared to our previous model through unreplicated blocking.

Earlier in this paper, we confirmed that x_4 was the most significant main effect - as a result, we will implement a split-block design (a special case of factorial design treatment) in order to evaluate the effects of the factor x_4 based on its correspondence to the output of strength. Remember, the purpose of this is so that we can suppress the influence of x_4 and observe the effect on the model adequacy and statistical significance.

Shown here is the code in which we designed the confounded blocking of x_4 in our new factorial experiment.

```
# Blocking confounded by X4 term
G <- rbind(c(X1 = 0, X2 = 0, X3 = 0, X4=1, X5=0))
design.1 = conf.design(G, p=2, block.name = "Block", treatment.names = c("X1", "X2","X3", "X4","X5"))
design.1$Y = ceramic$Y
```

`G` represents the term we will confound by, and we use the `conf.design()` function to create a dataset confounded by `G` and set the levels for each factor to 2.

We then formed a new model based on the blocks we made initially and the predictor variables x_1, x_2, x_3, and x_5 with their interaction terms, which would in turn reduce x_4's impact on the model since it is not included in the predictor variables. We then use our design instead of the original dataset so that we can confound properly.

```
> ceramic.3 = lm(Y~Block + X1*X2*X3*X5, data = design.1)
> summary(ceramic.3)

Call:
lm.default(formula = Y ~ Block + X1 * X2 * X3 * X5, data = design.1)

Residuals:
    Min      1Q  Median      3Q     Max
-26.810  -8.582   0.000   8.582  26.810

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    676.325     12.649  53.468  < 2e-16 ***
Block1         -64.859      6.136 -10.571 2.40e-08 ***
X11             27.745     17.355   1.599  0.13073
X21             12.450     17.355   0.717  0.48415
X31              0.535     17.355   0.031  0.97581
X51           -176.655     17.355 -10.179 3.95e-08 ***
X11:X21        -31.605     24.543  -1.288  0.21735
X11:X31        -58.020     24.543  -2.364  0.03199 *
X21:X31         -9.555     24.543  -0.389  0.70251
X11:X51        -40.020     24.543  -1.631  0.12379
X21:X51        -31.480     24.543  -1.283  0.21908
X31:X51        -49.360     24.543  -2.011  0.06263 .
X11:X21:X31     53.340     34.709   1.537  0.14517
X11:X21:X51    135.205     34.709   3.895  0.00143 **
X11:X31:X51     28.675     34.709   0.826  0.42166
X21:X31:X51     17.185     34.709   0.495  0.62769
X11:X21:X31:X51 -53.235    49.086  -1.085  0.29526
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.35 on 15 degrees of freedom
Multiple R-squared:  0.9884,    Adjusted R-squared:  0.9761
F-statistic: 80.16 on 16 and 15 DF,  p-value: 1.307e-11
```

We see that our adjusted R-squared for `ceramic.3` had a considerable increase to 0.9761. Some of the interaction terms and main effects have less significance, but in terms of how much the sample variation is measured through our model and the corresponding dataset, we are satisfied with this outcome.

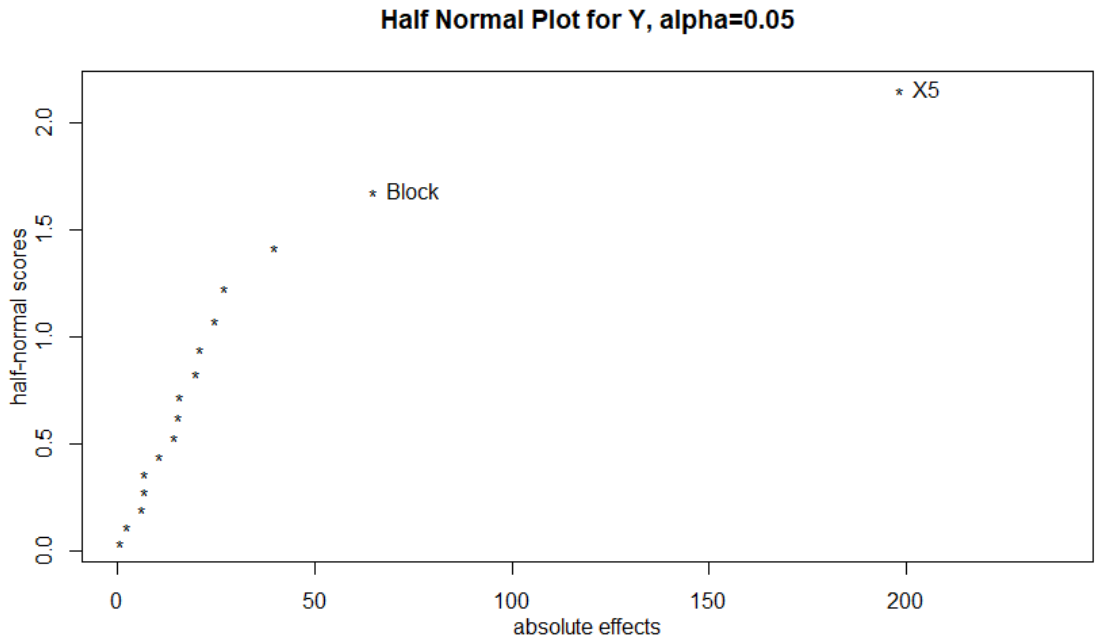
In addition to our improved adjusted R-squared value, the residual standard error of the `ceramic.3` is 17.35 on 15 d.f., which is lower than the residual standard error of `ceramic.2`, which is 24.3 on 23 d.f., indicating that the confounded model fits the data better than the unreplicated blocking design.

A simple `anova()` function call would explicate this thought process.

```
> anova(ceramic.3)
Analysis of Variance Table

Response: Y
            Df Sum Sq Mean Sq   F value    Pr(>F)
Block        1  33654   33654  111.7408 2.396e-08 ***
X1           1    894     894    2.9694 0.1053966
X2           1   3497    3497   11.6117 0.0038966 **
X3           1  12664   12664   42.0481 1.031e-05 ***
X5           1 315133  315133 1046.3327 2.730e-15 ***
X1:X2        1   4873    4873   16.1784 0.0011081 **
X1:X3        1   1839    1839    6.1052 0.0259539 *
X2:X3        1    307     307    1.0209 0.3283388
X1:X5        1   1637    1637    5.4360 0.0340792 *
X2:X5        1   1973    1973    6.5500 0.0217928 *
X3:X5        1   3158    3158   10.4866 0.0055139 **
X1:X2:X3     1    357     357    1.1855 0.2934243
X1:X2:X5     1   5896    5896   19.5752 0.0004923 ***
X1:X3:X5     1      2       2    0.0070 0.9342981
X2:X3:X5     1     44      44    0.1477 0.7061303
X1:X2:X3:X5  1    354     354    1.1762 0.2952553
Residuals   15   4518     301
---
```

Shown here, we have an additional slot of p-values, and we can observe that most of the variables that were selected seemed to be statistically significant in our anova function call. Not only that, it seems to boast a better model adequacy than what we initially expected from our lm summary. Again, it reaffirms the notion that we can be satisfied with our confounded design much more than our unreplicated blocking design.

Shown below is an additional Daniel half-normal plot of `ceramic.3`:



Now moving onto some of the methods we used to test residuals.

The residual that is most commonly seen is just the error which can be calculated (in this case) by subtracting our predicted mean ceramic strength from the value of our response from the corresponding values of the predictors.

Before continuing a matrix with interesting properties that is useful for examining residuals is the "Hat" matrix.
It can be found by using some matrix algebra:

$e = Y - \hat{Y}$- This is just our standard residual where e is our error matrix, Y is our response, and $\hat{Y}$ is our predicted.
$= Y - X\hat{\beta}$ Because our predicted can be found by multiplying our predictors and our Beta estimates.
Now let this be equal to $= Y - HY$
Doing some matrix algebra we have $H = X * (X^T X)^{-1} X^T$

This matrix is a square matrix with some interesting properties that are useful. For example, the diagonal elements have the property:
$Var(e_i)$(Variance of the error at point i) is equal to $\sigma^2(1 - h_{ii})$

And the sum of h_{ii} is equal to the number of predictors +1 for the intercept.

Now getting back to our different types of residuals.

A slightly modified version of residual is the standardized residual which can be found by dividing the error by the MSE, this is essentially the z-score for residual. A deleted residual, denoted d_i, is the difference between the observed response y_i and the predicted value $y\hat{}_{(i)}$ when the data from the ith observation is deleted from the analysis. An observation with large deleted residual indicates that the point may have a large influence on the fitted model.

The last type of residual we will be mentioning is the R-Student statistic which is the standardized version of the deleted residual and is also known as the externally studentized residual.

Now, looking at the detection of influential points there are two methods I will be discussing the first one is Cook'D and the second one is DFFITS.

Cook'D is a measure of the influence of an observation on the beta coefficients. The Cook's Distance for observation i, D_i, is
calculated as follows:

$$D_i = \frac{(y_i - \hat{y}_i)^2}{(k+1)\text{MSE}} \left[\frac{h_i}{(1-h_i)^2} \right]$$

-From package olsrr Measures of Influence (r-project.org)

The final measure of influence I will be mentioning is DFFITS. DFFITS is calculated by the difference between the predicted, $y\hat{}_i$, and the predicted with the ith observation deleted,$y\hat{}_{(i)}$. Using package olsrr a observation is determined influential if the DFFITS value is greater than the following equation:

$$2 * \frac{\sqrt{(p+1)}}{(n-p-1)}$$

-From package olsrr Measures of Influence (r-project.org)

- where n is the number of observations and p is the number of predictors including the intercept.

Results

It almost seems as if it completely satisfies the assumption of normality as well as the assumption of constant variances. Another check to confirm this would be to conduct a Shapiro-Wilk normality test and see if the graph lines up with our belief that it is truly normal. Just as a model comparison, we will conduct another Shapiro-Wilk normality test for our unreplicated blocking design just to compare both our models.

```
> shapiro.test(ceramic.3$residuals)

        Shapiro-Wilk normality test

data:  ceramic.3$residuals
W = 0.99477, p-value = 0.9999

> shapiro.test(ceramic.2$residuals)

        Shapiro-Wilk normality test

data:  ceramic.2$residuals
W = 0.96556, p-value = 0.3866
```

As shown above, we see that the first Shapiro-Wilk test of normality is for the residuals of `ceramic.3`, our confounded model. The second Shapiro-Wilk test of normality is for the residuals of `ceramic.2`, the model obtained via unreplicated blocking. We see that it produces a p-value of 0.9999, meaning that we can reject the null hypothesis completely. Thus, it shows that it is completely normal and that there is no statistical departure from normality in the slightest.

However, when we compare the results of the first Shapiro-Wilk test to that of the unreplicated blocking design, although it was also reasonably normal in the model's residuals, the p-value of the Shapiro-Wilk test for the unreplicated blocking design does not come close to the p-value of our confounded model, encouraging us to use the confounded model.

While our model analysis is complete, we still must factor in how effective our model will be when testing new data. We might run into the danger of overfitting our model to every single data point and shifting our model due to the "noise" it produces as a result. Overall, we want to obtain a model that uses its initial dataset to not only predict the response variable strength, but to also be a good indicator of the strength of the grinding parameter based on any of the given predictor variables that we are concerned with using in the future.

Now, to make sure that we do not factor in the noise for any new information that we obtain for this specific ceramic dataset, we will proceed with 10-fold cross-validation to prevent overfitting and to bolster model adequacy.

We run this cross-validation in accordance with our confounded block design versus our unreplicated block design to compare which would produce more satisfactory results.

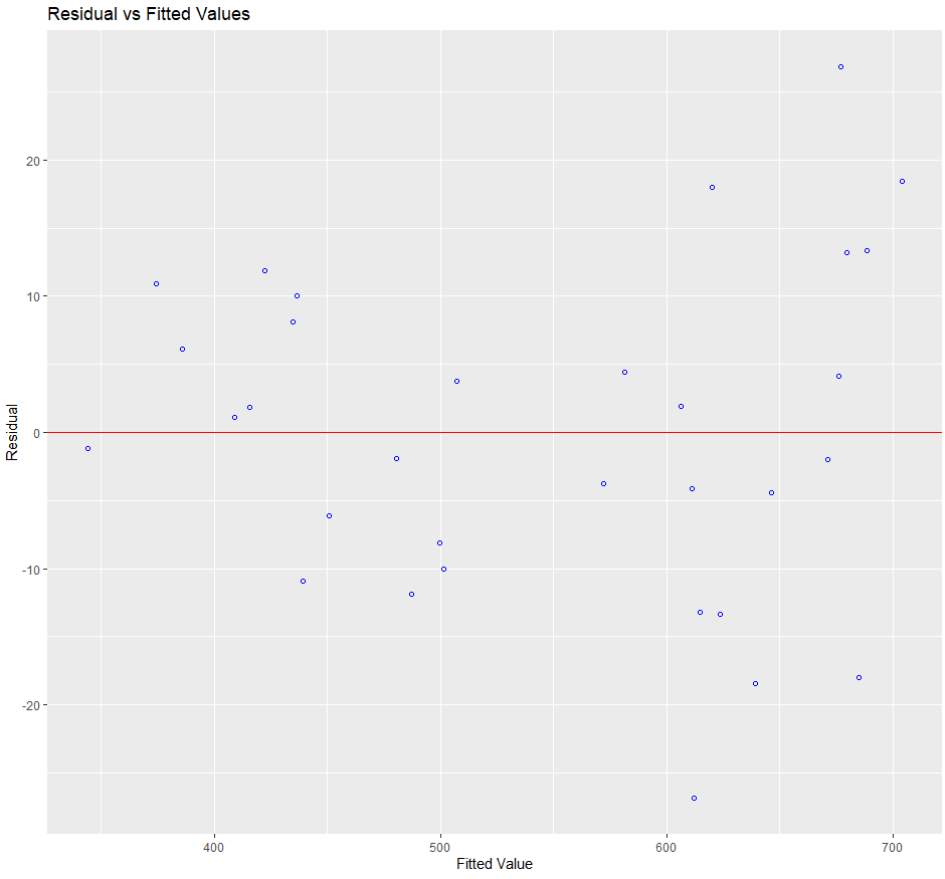
```
> set.seed(666)
> train.control <- trainControl(method = "cv", number = 10)
> modelnew2 <- train(Y~X1+X2+X3+X5+X4+X1:X2+X1:X2:X4 + Block, data = ceramic, method = "lm",
+                    trControl = train.control)
> print(modelnew2)
Linear Regression

32 samples
 6 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 28, 30, 30, 28, 29, 28, ...
Resampling results:

  RMSE      Rsquared   MAE
  24.97633  0.9874571  22.04096

Tuning parameter 'intercept' was held constant at a value of TRUE
>
```

```
> train.control <- trainControl(method = "cv", number = 10)
> modelnew3 <- train(Y~Block + X1*X2*X3*X5, data = design.1, method = "lm",
+                    trControl = train.control)
> print(modelnew3)
Linear Regression

32 samples
 5 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 29, 28, 29, 29, 29, 29, ...
Resampling results:

  RMSE      Rsquared   MAE
  24.16373  0.9709426  20.94193

Tuning parameter 'intercept' was held constant at a value of TRUE
```

Shown here are the results of the 10-fold cross validation on the unreplicated block design and the confounded block design respectively. A quick comparison of both the models shows us that our unreplicated block design produces a better cross-validation in accordance with the R-squared value, 0.987 versus 0.971. Ostensibly, it seems to be that we would want to use our unreplicated block design to be our final model. However, let us look at all the other information. We see that in our confounded block design, we would see that the R-squared is still very good, only a 0.1 difference from the other model's results. Moreover, if we compare the Root Mean Square Error and the Mean Absolute Error slots for both of the summary analyses, we

can observe that our confounded block design shows more credence and would be more optimal to use as our final model. We want our errors to be as small as possible when gathering and fitting data to our model. All in all, we would still proceed with using our confounded block design above our unreplicated block design as our final model.

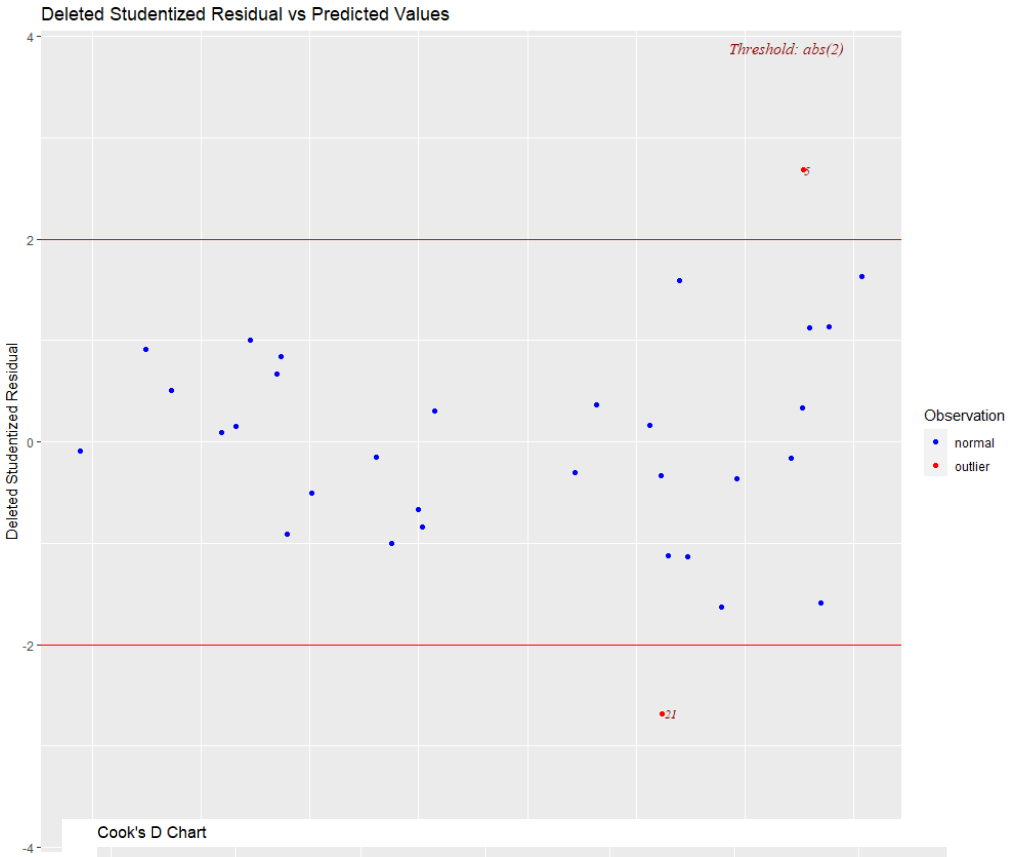
Now continuing with a deeper level of residual analysis on our final model, the confounded block design (`ceramic.3`), let us first examine the residual plot: made using ols_plot_resid_fit from the olsrr package:



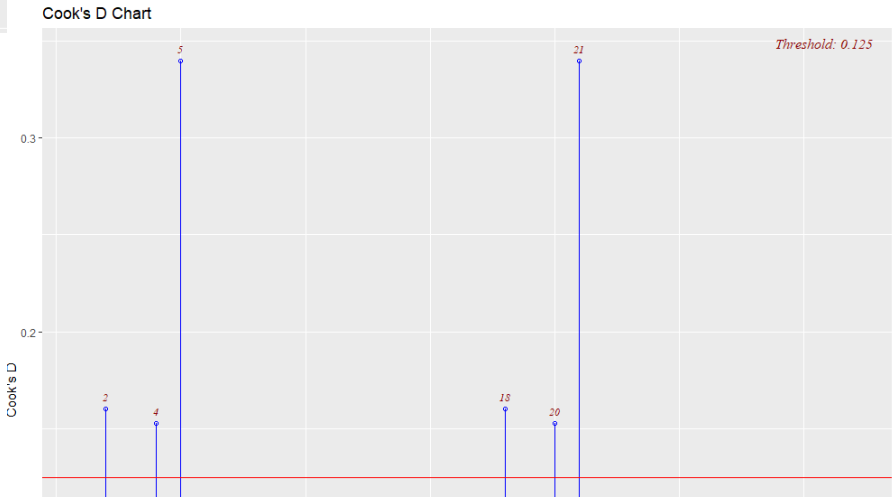
Assumptions of homoscedasticity appear to be reasonably satisfied although there appears to be a few outliers.
The standardized and deleted residual plots may give us a clue in determining if they are influential:

*made using the ols_plot_resid_stand and the ols_plot_resid_stud_fit

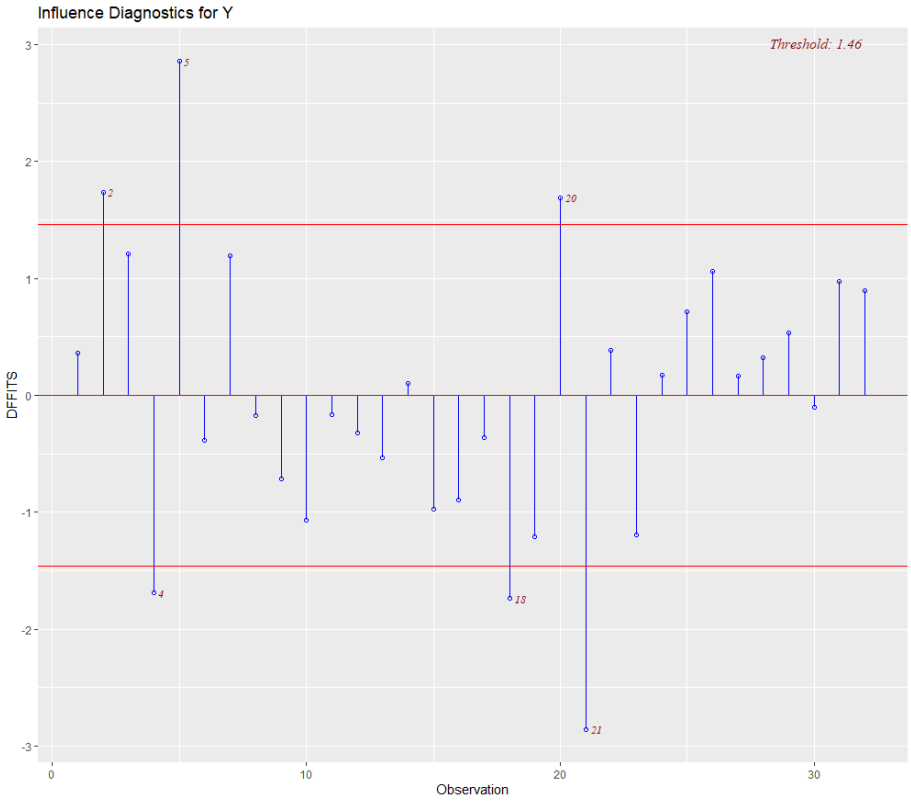
We can see that our standardized residuals chart indicates that points 5 and 21 may be outliers. Let us see what our deleted residual plot indicates:



As seen here, the deleted residuals also indicated that points 5 and 21 are outliers, now let us look at our Cook'D plot and DFFITS to check if they also have high influence:



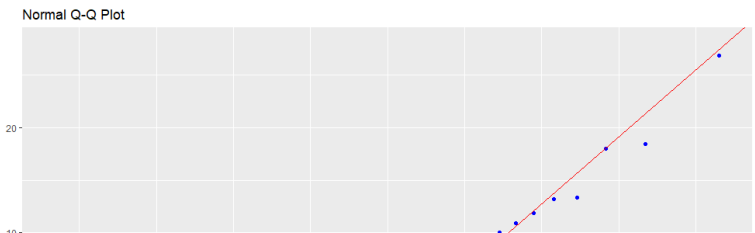
As we can see, both points 5 and 21 are points with high influence and now looking at our DFFITS plot:



Now that we have seen a few high influence points in our model, let us examine what levels of our factors these points correspond to.

First examining our model matrix point 5 corresponds to our block, and every other factor at a low level other than grit and observation 21 corresponds to the block at a high level and grit at a high level with the rest of the factors at the low level. Which indicates that in both blocks grit alone at the high level is influential.

Lastly, let us check for the normality of the errors using the ols_plot_resid_qq function from package olsrr:



As the points appear to lie linearly on the line the assumption of normality of errors is satisfied.

Discussion

Last, but not least, let us compare our `ceramic.3` model to the model obtained by NIST's experiment.

Shown below is their original model, which contained only the main effects and the significant interaction terms:

```
> #11-effect model from NIST
> ceramicNISTdata = read.delim("CeramicData.txt", sep = "|")
> ceramic.NIST = lm(Y~X1+X2+X1*X2+X3+X1*X3+X4+X1*X4+X2:X4+X1:X2:X4+X3*X4+X5 + X4:X5, data =
 ceramicNISTdata)
> summary(ceramic.NIST)

Call:
lm.default(formula = Y ~ X1 + X2 + X1 * X2 + X3 + X1 * X3 + X4 +
    X1 * X4 + X2:X4 + X1:X2:X4 + X3 * X4 + X5 + X4:X5, data = ceramicNISTdata)

Residuals:
    Min      1Q  Median      3Q     Max
-35.118  -4.817  -0.321   5.833  23.170

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  546.896      2.645 206.751  < 2e-16 ***
X1             5.287      2.645   1.999 0.060170 .
X2            10.454      2.645   3.952 0.000855 ***
X3           -19.893      2.645  -7.521 4.14e-07 ***
X4           -99.237      2.645 -37.516  < 2e-16 ***
X5           -32.430      2.645 -12.260 1.80e-10 ***
X1:X2         12.340      2.645   4.665 0.000169 ***
X1:X3         -7.580      2.645  -2.866 0.009896 **
X1:X4          7.153      2.645   2.704 0.014065 *
X2:X4          7.852      2.645   2.968 0.007897 **
X3:X4         -9.935      2.645  -3.756 0.001338 **
X4:X5          6.444      2.645   2.436 0.024859 *
X1:X2:X4      13.573      2.645   5.131 5.93e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.96 on 19 degrees of freedom
Multiple R-squared:  0.9891,    Adjusted R-squared:  0.9822
F-statistic: 143.9 on 12 and 19 DF,  p-value: 5.248e-16
```

We have the adjusted R-squared value at 0.9822, however, the NIST experiment did not stop here. They implemented a Box-Cox Transformation for the ceramic strength, which is the response variable, and they obtained an optimal lambda of 0.2.

They then transformed their response data using the following equation which we then rewrote in R:

$$\frac{(y_i)^{\lambda} - 1}{\lambda \left[\left(\prod_{i-1}^{n} y_i \right)^{\frac{1}{n}} \right]^{\lambda - 1}}$$

```
#transformed data
cer.y = as.vector(ceramicNISTdata$Y)
x = prod(cer.y)
x = x^(1/32)
x = x^(0.2-1)
denom = 0.2*x
ceram.y = ceramicNISTdata$Y^0.2
num = ceram.y-1
ceramicNISTdata$Y = num/denom
```

In their final model, they dropped the x_4:x_5 interaction term. We will also do the same.

Their final model will now look like this: (they dropped the X4:X5 interaction term)

```
> #final model
> ceramic.NIST.2 = lm(Y~X1+X2+X1*X2+X3+X1*X3+X4+X1*X4+X2:X4+X1:X2:X4+X3*X4+X5, data =
 ceramicNISTdata)
> summary(ceramic.NIST.2)

Call:
lm.default(formula = Y ~ X1 + X2 + X1 * X2 + X3 + X1 * X3 + X4 +
    X1 * X4 + X2:X4 + X1:X2:X4 + X3 * X4 + X5, data = ceramicNISTdata)

Residuals:
    Min      1Q  Median      3Q     Max
-29.468  -3.717  -1.389   6.560  20.294

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1917.115      2.441 785.252  < 2e-16 ***
X1             5.777      2.441   2.366 0.028183 *
X2            11.691      2.441   4.789 0.000112 ***
X3           -21.649      2.441  -8.867 2.29e-08 ***
X4           -99.272      2.441 -40.662  < 2e-16 ***
X5           -31.871      2.441 -13.055 3.03e-11 ***
X1:X2         14.467      2.441   5.926 8.53e-06 ***
X1:X3         -7.339      2.441  -3.006 0.006979 **
X1:X4          7.189      2.441   2.944 0.008016 **
X2:X4          9.160      2.441   3.752 0.001255 **
X3:X4        -12.965      2.441  -5.311 3.38e-05 ***
X1:X2:X4      15.325      2.441   6.277 3.96e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.81 on 20 degrees of freedom
Multiple R-squared:  0.9904,    Adjusted R-squared:  0.9851
F-statistic: 187.8 on 11 and 20 DF,  p-value: < 2.2e-16
```

At a first glance, we see that their model improved from an adjusted R-squared of 0.9822 to 0.9851. Comparing their improved model to our `ceramic.3`, we can see that it has a higher adjusted R-squared value than ceramic.3 as well as a lower residual standard error, indicating that their model may indeed have a significantly better fit. However, we cannot assume this yet. Let us use 10 fold cross validation to further analyze this.

```
> set.seed(666)
> train.control <- trainControl(method = "cv", number = 10)
> modelnew <- train(Y~X1+X2+X1*X2+X3+X1*X3+X4+X1*X4+X2:X4+X1:X2:X4+X3*X4+X5
 data = ceramicNISTdata, method = "lm",
+                   trControl = train.control)
> print(modelnew)
Linear Regression

32 samples
 5 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 28, 30, 30, 28, 29, 28, ...
Resampling results:

  RMSE      Rsquared   MAE
  16.80562  0.9907078  13.85643

Tuning parameter 'intercept' was held constant at a value of TRUE
> train.control <- trainControl(method = "cv", number = 10)
> modelnew2 <- train(Y~Block + X1*X2*X3*X5, data = design.1, method = "lm",
+                   trControl = train.control)
> print(modelnew2)
Linear Regression

32 samples
 5 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 29, 28, 29, 29, 29, 29, ...
Resampling results:

  RMSE      Rsquared   MAE
  24.16373  0.9709426  20.94193

Tuning parameter 'intercept' was held constant at a value of TRUE
```

The 10 fold cross validation shows us that the model obtained by NIST after transforming the response variable had a better R-squared, RMSE, and MAE than our model. We can conclude that their model is better than what we obtained in our model.

To summarize, we initially arrived at the conclusion that the model obtained through performing an unreplicated block design (`ceramic.2`), which was still a very good model, was not as effective as the one obtained by the confounded block design (`ceramic.3`). The Shapiro-Wilk test for normality of the residuals, the adjusted R-squared values, and the 10 fold cross validation all proved this to be true. The

purpose of this experiment was to determine which factors and interaction terms are significant in influencing the strength of the ceramic when exposed to those grinding parameters.

However, compared to the NIST transformed model, we can conclude that blocking and confounding, while still highly effective, wasn't the best approach we could use for this experiment. This could be due to the fact that our approach could have overfitted, another possibility could be that confounding a significant main effect would hurt the overall model adequacy.

Acknowledgements

Class material provided by Professor Jack Mardekian, Rutgers University, and the dataset provided by NIST.

Literature Cited

https://en.wikipedia.org/wiki/Ceramic

NIST Experiment:
https://www.itl.nist.gov/div898/handbook/pri/section5/pri598.htm

https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/doe/how-to/response-surface/analyze-response-surface-design/interpret-the-results/all-statistics-and-graphs/effects-plots/#:~:text=The%20normal%20probability%20plot%20of%20the%20effects%20shows%20the%20standardized,that%20the%20effect%20is%200.
https://stat.ethz.ch/education/semesters/as2015/anova/08_Split_Plots.pdf

https://stat.ethz.ch/education/semesters/as2015/anova/08_Split_Plots.pdf

William Mendenhall and Terry Sincich, *A Second Course IN Statistics: Regression Analysis seventh edition.*

Package Olsrr documentation: olsrr.pdf (r-project.org)

Contributions

Ritwik served as Project Coordinator and performed the unreplicated blocking and confounding in RStudio, the rewriting of the NIST experiment in R, the 10 fold cross validation of the NIST transformed model, as well as writing the Abstract and Discussion sections. Rishab was responsible for the write-up of the Introduction, Materials & Methods, and Results sections. Ben was responsible for the 10-fold cross validation of both blocking and confounding models in RStudio and coming up with exam questions. Karthik performed the residual analysis and was responsible for the write up of the Discussion section. All group members contributed equally and sufficiently to ensure the quality and completion of this project.

Sample Exam Questions

1) Choose the appropriate multiple choice answer.

Treatment Combination	Factorial Effect								
	I	*A*	*B*	*C*	*AB*	*AC*	*BC*	*ABC*	Block
(1)	+	−	−	−	+	+	+	□	□
a	+	+	−	−	−	−	+	□	□
b	+	−	+	−	□	□	□	□	□
ab	+	+	□	−	+	−	−	□	□
c	+	□	−	+	+	−	□	□	□
ac	+	+	−	□	−	□	−	□	□
bc	+	−	+	+	−	−	+	□	□
abc	+	+	+	+	□	+	+	□	□

Answers go from left to right:

a. Which of the following choices can replace the blanks for treatment b?
   i. −, −, −, −, 1
   ii. −, −, +, −, 2
   iii. −, +, − , +, 1
   iv. −, +, − , +, 2
b. Which of the following choices can replace the blanks for treatment c?
   i. −, −, +, 2
   ii. −, −, +, 1
   iii. +, −, +, 2
   iv. −, −, −, 2
c. Which of the following choices can replace the blanks for treatment bc?
   i. −, 1
   ii. +, 1
   iii. +, 2
   iv. −, 2

solution

Treatment Combination	Factorial Effect								
	I	*A*	*B*	*C*	*AB*	*AC*	*BC*	*ABC*	Block
(1)	+	−	−	−	+	+	+	−	1
a	+	+	−	−	−	−	+	+	2
b	+	−	+	−	−	+	−	+	2
ab	+	+	+	−	+	−	−	−	1
c	+	−	−	+	+	−	−	+	2
ac	+	+	−	+	−	+	−	−	1
bc	+	−	+	+	−	−	+	−	1
abc	+	+	+	+	+	+	+	+	2

M/C answers: iv, i, i

2) Short answer

You started collecting data for an experiment for a new drug to test its effect on factors that impact mental health. You began collecting data in early 2020 Your experiment is blocked for age, sex and a placebo group. You continued collecting data through 2020. It's time to start analyzing your data and you are wondering if the effect of Covid 19 is a confounding factor in your data and if you should start collecting data again once more people have been vaccinated and the pandemic is winding down.. What is your conclusion? *more than one acceptable answer

Possible solution:
One case could be: the effects of Covid 19 have been randomly distributed throughout all of the participants. We can measure the effects of the pandemic as a whole and control for it.

3) True Or False

Let us say we have Model 1 and Model 2 for linear regression. Model 2 had a higher R^2 than Model 1.
Higher R^2 will always mean that it is a better model. (Provide a reason if false)

Solution:
False, just because we have a higher R^2 does not indicate that all the time it will be a better model. We would want to look at residuals and see if they satisfy normality and constant variances. Also, we want to see RMSE and MAE just to make sure that the model would not overfit as well as provide misinformation for any out of sample/future information.