How to Prevent Failures via an Augmented Screening Experiment

Experiments on machine-made bread show how the clever application of two-level factorial design of experiments (DOE) quickly screens inputs to see which ones might cause a system to fall flat. When the process fails, the first DOE is augmented with additional runs that reveal harmful interactions of input factors.

by Mark J. Anderson

Introduction

Failures in processing and product application can often be prevented by applying a form of design of experiments called "ruggedness testing." Ruggedness testing challenges your process, product or method to discover how outputs change as variables fluctuate over ranges encountered during normal use. Originally proposed for quality assurance on analytical methods,[1] ruggedness testing is now used for validation of medical devices[2] and other manufacturing processes.

Ruggedness tests typically include as many factors, each tested only at two levels, in as few runs as possible, for example, 7 factors in 8 runs.[3] These are called "saturated" two-level fractional factorial designs. Other examples of saturated designs are 11 factors in 12 runs and 15 factors in 16 runs. The efficiency of such designs comes at a price—they provide low resolution due to aliasing of main effects with two-factor interactions. This won't matter in the ideal case where statistical analysis of variance (ANOVA) reveals no significant effects. Then the system passes the ruggedness test and it can be released with no further experimentation. However, if ANOVA comes out significant, the system fails the test and it must be reengineered. Further experimentation, via augmentation to the original DOE, then must be performed to resolve the true effects causing the failure.

The big assumption in ruggedness testing is that none of the factors will be significant. It may be more realistic to expect that some factors, thought to be insignificant, will create a discernible effect on the system and thus be a potential cause for failure. With this in mind, it makes no sense to employ low resolution saturated designs, which can be likened to kicking your PC (or slapping the monitor) to make it work. Obviously this crude approach to problem-solving won't reveal the true causes for failure. On the other hand, doing a high resolution (or full) factorial design with many runs could be very wasteful if it turns out that the process actually is rugged; in other words, none of the factors affect it significantly. What's needed is

a medium resolution "screening" design that strikes a good compromise between the investment in experimental runs and the return of information on the effects.

This article details an application of screening design that anyone can relate to—baking bread in a home machine. The bread DOE investigated alternative ingredients to see which, if any, would cause the machine to fail. It demonstrates the benefits of fractional factorial design and illustrates the often misunderstood impacts of aliasing on resolution of effects. Then it shows how to resolve aliased interactions via a technique for design augmentation called "semifold." The ideas presented here are meant to be applied to all manufacturing processes and not restricted only to the kitchen. The case study on bread-making will be presented in a light-hearted manner, but the implications of testing for ruggedness should be taken seriously by anyone responsible for manufacturing quality.

Case study—sifting through flour and other ingredients for bread

Modern bread-making machines make it easy to bake your own bread. For years I produced machine-made bread using pre-mixes, but I decided that it would be more economical (and fun) to buy the ingredients separately and do some baking experiments. I wanted to see if I could save money by using the regular (and cheaper) varieties of flour and/or yeast, as opposed to those that were specifically advertised for bread-making machines. Also, according to a recipe I found printed on the bread-flour package, I could economize by using margarine and water versus butter and milk as fluids. I assumed that any combination of these ingredients would work, and that none of my family members would notice. To be sure, I set up a screening design on the four main ingredients (arbitrarily coded minus (–) and plus (+), respectively):

A. Liquid: Water (–) or Milk (+)
B. Oil: Butter (–) or Margarine (+)
C. Flour: Regular (–) or Bread (+)
D. Yeast: Regular (–) or Bread (+)

Baking the 16 loaves required for the full factorial (2x2x2x2 or 2^4) would be wasteful, especially if nothing perceptibly changed, so I chose a standard half-fraction requiring only 8 runs. [Warning: Before running such a small experiment, check its power. If inadequate, abandon the fraction and go back to the full factorial.] A design like this works well if nothing comes out significant, or you see only main effects of the test factors, but you lose resolution on interactions of factors (more on this later!). It boils down to a trade-off of experimental runs versus information. In this case I thought none of the factors would be significant so it made sense to choose the lower-resolution half-fraction design.

For each production run I added the liquid, oil, flour and yeast at the specified levels, as well as a constant amount of salt and sugar. Processing factors such as time, temperature and mixing were kept as constant as possible by always using the same setting on the bread-making machine. I measured the taste of the resulting

loaves by averaging ratings, scaled low to high from 1 to 10, from members of my household. However, this turned out to be a waste of time because surprisingly many of my loaves failed to rise, thus making them inedible. Figure 1 shows a picture of one of my failed loaves. It wasn't pretty!

Figure 1. Author Inspects Failed Bread

At this point my focus shifted to finding the cause for failure. (For what it's worth, I detected no significant difference in the taste ratings of loaves that did rise.) I observed that the bread either rose or fell flat, there was really no in-between, so I simply coded them 1 (good) or 0 (bad); respectively. In hindsight, actual measurements of bread height would have led to greater precision in statistical analysis, but this binary scale for pass/fail got the job done and made it easier to see what happened.

Table 1 shows the results of the screening design on bread-making. The runs are listed in standard order but I actually performed them at random to avoid bias due to ongoing machine wear, aging of the ingredients, etc.

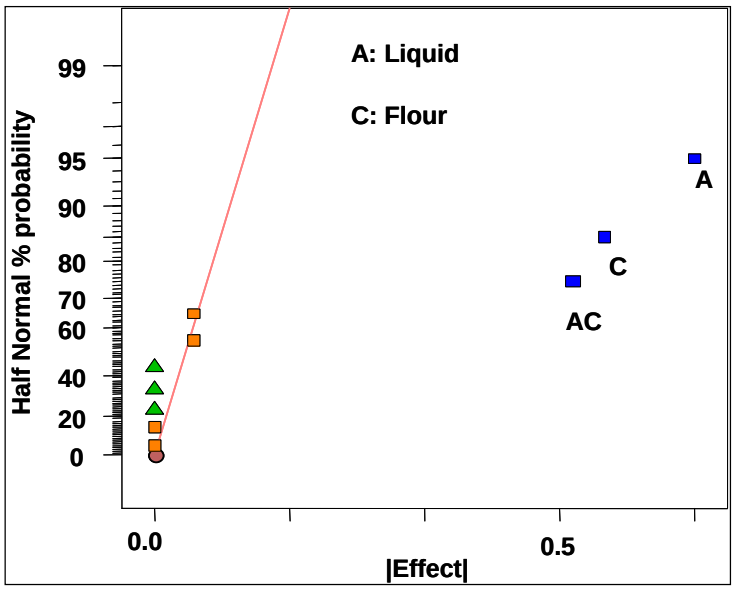
Table 1. Screening design on bread-making

Std	A:Liquid	B:Oil	C:Flour	D:Yeast	Rise
1a,b	Water	Butter	Regular	Regular	0, 0
2a,b	Milk	Butter	Regular	Bread	1, 1
3a,b	Water	Margarine	Regular	Bread	0, 0
4	Milk	Margarine	Regular	Regular	1
5	Water	Butter	Bread	Bread	1
6	Milk	Butter	Bread	Regular	1
7	Water	Margarine	Bread	Regular	1
8	Milk	Margarine	Bread	Bread	1

Notice that I repeated the two runs that failed (1a,1b and 3a,3b) and one that didn't (2a, 2b). I got the same results, bad and good. This proved that the results could be reliably reproduced.

Which of the tested ingredients, if any, caused the failure of the bread to rise? A person might find the answer by inspection, but it's quicker to let DOE software[4] tell the story via a half-normal probability plot of the effects (see Figure 2).

Figure 2. Half-Normal Plot of Effects Reveals Interaction



The x-axis on this plot shows the absolute value of the effects. It's laid out in the scale of 0 to 1 that I used to quantify success or failure of the bread-making process. The squares represent the seven effects that can be estimated from the eight unique combinations of ingredients included in the ruggedness test. The three triangles come from the pure error estimate obtained by replicating combinations 1, 2 and 3. Notice how these estimates of error (triangles), as well as four of the effects, fall on a line near the zero-effect level. (Due to the 0 or 1 scale of measurement, the pure error came out exactly at zero.) The y-axis on the half-normal plot is scaled in a special way to make normally distributed effects line up in this fashion.[5] Effects that fall far from the line, in this case liquid (A), flour (C) and their interaction (AC), can be assumed to be significantly large relative to what might be expected from normal variation.

However, things are not as clear-cut as they appear from the plot: Due to the nature of fractional design, interaction AC is *aliased* with BD. This is the price you pay by cutting out half the runs. Let's see what it means to be aliased and how this affects the resolution of effects.

The pitfalls of doing fractional two-level factorial design

Take a look at Table 2, which lays out the experimental design in coded levels with all interaction columns included. The factor coding is simple: minus (–) for one level versus plus (+) for the other. For numerical factors such as time or temperature, these codes would be assigned to the low and high levels,

respectively, but in this case the factors are categorical, so assignment of minuses and pluses is arbitrary (I used cost as the key). The coding for interaction columns is done by multiplying parent terms. For example, the AC column is computed by multiplying the A by the C column.

Table 2. Test Layout in Coded Levels with Interactions Shown

Std	A	B	C	D	AB	AC	AD	BC	BD	CD	ABC	Rise
1a,b	–	–	–	–	+	+	+	+	+	+	–	0, 0
2a,b	+	–	–	+	–	–	+	+	–	–	+	1, 1
3a,b	–	+	–	+	–	+	–	–	+	–	+	0, 0
4	+	+	–	–	+	–	–	–	–	+	–	1
5	–	–	+	+	+	–	–	–	–	+	+	1
6	+	–	+	–	–	+	–	–	+	–	–	1
7	–	+	+	–	–	–	+	+	–	–	–	1
8	+	+	+	+	+	+	+	+	+	+	+	1

Observe that, due to the way this matrix is laid out, A times C (column AC) equals B times D (column BD). These two interactions are therefore statistically "aliased." It's impossible to say which one is really causing the significant effect on bread-making performance because they change back and forth from one level to the other in exactly the same pattern (see italicized columns in the table). Upon closer inspection, you'll notice that all of the two factor interactions are aliased: AB = CD and AD = BC. It also turns out that all main effects get aliased with a three-factor interaction (for example: A = BCD, not shown in the table above, but easily computed by multiplying BxCxD). It's a generally acceptable practice to ignore interactions of three or more factors such as BCD, because they're so improbable in most systems. This assumption becomes more of a judgment call in chemical processes such as food-baking. It's always a good idea to consult a subject matter expert on what factors might interact and to what degree.

Statisticians characterize the level of aliasing in the design chosen for bread-making as "resolution IV." To help you grasp the concept of resolution, think of main effects as 1 factor and add this to the number of interacting factors it will be aliased with. Resolution IV indicates a 1-to-3 (example: A=BCD) and 2-to-2 aliasing (AB=CD, etc.), both of which add to 4 (2+2 and 1+3, respectively). It's possible to create resolution III designs, for example by testing 7 factors in only 8 runs. Then you'd alias main effects with two-factor interactions (1+2=3), which wouldn't be good. If you can afford more runs, choose a design with at least a resolution V, such as 5 factors in 16 runs. Then main effects get aliased only with extremely unlikely interactions of four-factors (1+4=5) and two-factor interactions are confused only with three-factor interactions (2+3=5). However, a resolution IV design often offers a reasonable compromise for testing factors that are thought unlikely to affect performance, which is what I expected when I set up my screening

design on bread-making. The introduction to this article referred to low, medium and high degrees of resolution which now can be defined as Resolution III, IV and V or better; respectively. To interpret design resolution it may help to think in terms of colors on a stoplight:

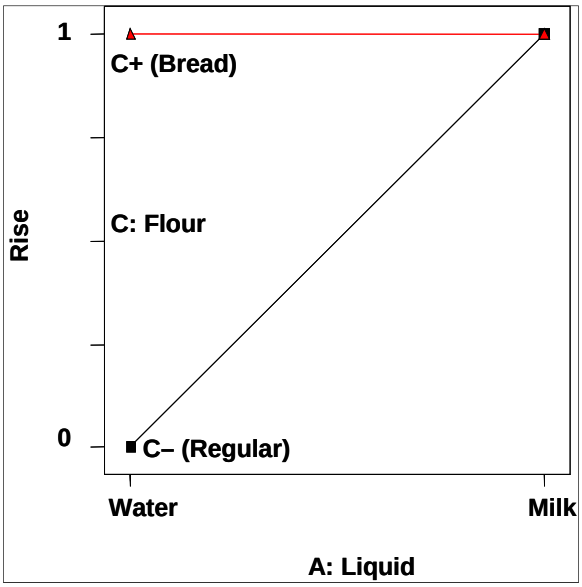
Red for low resolution III (stop and think before proceeding)

Yellow for medium resolution IV (proceed with caution)

Green for high resolution V or better (go ahead).

If the concept of resolution remains a bit fuzzy, do not belabor it. In the end, what is really important are the specific aliases created by doing a fractional factorial. These can be found in textbooks[6] or generated by DOE software. In this case the only alias of concern was AC=BD, but since neither of the parents (B and D) of the BD interaction came out statistically significant, and these factors (oil and yeast) seemed unlikely to interact, I was tempted to take a leap of faith and place my bets on the AC interaction (Figure 3, shown below, with triangles representing bread flour and squares symbolizing the regular flour).

Figure 3. Interaction of Liquid (A) with Flour (C) Shows Apparent Problem with Rising



The combination of water and regular flour appeared to create problems with the bread-maker (zero rise at these conditions), but not being a gambling man, I felt it would be best to perform follow-up runs to separate this interaction effect (AC) from its alias BD. To resolve this problem, I made use of a nifty DOE method called a "semifold," which requires adding only half the runs of the original design.[7] This augmentation method was developed specifically to improve resolution of two-level fractional factorial designs, such as that used for bread-making, with aliased two-factor interactions (2fi's).

Semifolding the bread DOE to resolve aliased interactions

Before getting into the details of semifoldover, let's go back a step and talk about full foldover—an established method for enhancing resolution III designs. It's very simple to perform: Just repeat the original experiment with all factors at opposite levels. For details on a successful application of a complete foldover see the in-line skate example in my presentation to ASQ's Annual Quality Congress that formed the basis for this article.[8] You'd think that what's good for a resolution III design (doing a complete foldover) would be good for a resolution IV design, but that's not always the case. For example, see Table 3, which shows a complete foldover on the initial bread-making DOE. Notice that the second block of runs goes opposite the first on all levels (run 9 versus run 1, 10 versus 2, and so on.)

Table 3. Complete Foldover on Initial Bread-Making DOE

Std	Block	A:Liquid	B:Oil	C:Flour	D:Yeast
1	1	Water	Butter	Regular	Regular
2	1	Milk	Butter	Regular	Bread
3	1	Water	Margarine	Regular	Bread
4	1	Milk	Margarine	Regular	Regular
5	1	Water	Butter	Bread	Bread
6	1	Milk	Butter	Bread	Regular
7	1	Water	Margarine	Bread	Regular
8	1	Milk	Margarine	Bread	Bread
9	2	Milk	Margarine	Bread	Bread
10	2	Water	Margarine	Bread	Regular
11	2	Milk	Butter	Bread	Regular
12	2	Water	Butter	Bread	Bread
13	2	Milk	Margarine	Regular	Regular
14	2	Water	Margarine	Regular	Bread
15	2	Milk	Butter	Regular	Bread
16	2	Water	Butter	Regular	Regular

Did this really accomplish anything? It would do so only by creating additional unique combinations of factors. Unfortunately in this case, the complete foldover only replicates the existing runs. To see this, pair up run 8 with 9 and then move out from this center portion of the table to see the other replicates: 7-10, 6-11, 5-12, 4-13, 3-14, 2-15 and 1-16. To avoid failures like this in applying a complete foldover to resolution IV designs, you can do something a bit different—a single-factor foldover. This is easy to do—simply change

the signs only on one factor, while leaving all the other factors at their original levels. To make this work, you should single out a factor that's involved in the largest significant two-factor interaction (2fi) that's aliased with other 2fi(s). In the bread-making case, any one of the factors could be chosen because they're all involved in the AC-BD alias that must be resolved. I arbitrarily chose factor C (flour) as the single column to fold over (see Table 4). Notice how the levels on C-only go opposite from the original block of runs 1-8 shown in Tables 1 and 3 (run 9 versus run 1, 10 versus 2, and so on).

However, to accomplish the de-aliasing you actually need only perform half of these additional runs that result from the single-factor foldover. This refinement on the procedure is called a "semifold." It requires that you:

1. Lay out a single-factor foldover from the original design.
2. Perform only half of the foldover runs by selecting those where the chosen factor is either at its low level or high level, whichever you believe will generate the most desirable response(s).

I ran only the combinations calling for regular flour. Why? Because I preferred not to spend the money on the more expensive flour formulated specifically for bread.

Table 4. Single-factor Foldover on Bread-Making Experiment (Second Block Only)

Std	A:Liquid	B:Oil	C:Flour	D:Yeast	Rise
9	Water	Butter	Bread	Regular	
10	Milk	Butter	Bread	Bread	
11	Water	Margarine	Bread	Bread	
12	Milk	Margarine	Bread	Regular	
13	**Water**	Butter	**Regular**	Bread	**0**
14	Milk	Butter	Regular	Regular	1
15	**Water**	Margarine	**Regular**	Regular	**0**
16	Milk	Margarine	Regular	Bread	1

Now it could be seen that the combination of water and regular flour caused the bread-making to fail (zero rise). As shown in Table 5, this finding is unequivocal because the four-run semifold de-aliased the interaction of factors A and C from that of B and D. As you can see, the patterns no longer match.

Table 5. Second Bread-Making DOE: Design Layout in Coded Levels with Interactions Shown

Std	A	B	C	D	AB	AC	AD	BC	BD	CD	Rise
13	−	−	−	+	+	+	−	+	−	−	0
14	+	−	−	−	−	−	−	+	+	+	1

15	−	+	−	−	−	+	+	−	−	+	0
16	+	+	−	+	+	−	+	−	+	−	1

Therefore, I concluded that the interaction of factors A and C, depicted in Figure 2, accurately described what affected the bread-making process. You can see in Table 2 and Table 5 that whenever AC is at the plus level (the combination of water with regular flour), the bread failed to rise (score of 0). It's now obvious that I must avoid this particular combination of ingredients. That's not a problem, because with a family like mine, there's always milk in the refrigerator, so I just use it instead of water and the bread always rises. I use margarine and regular yeast with the regular flour to keep ingredient costs to a minimum.

Conclusions

If you are doing a DOE to screen potential variables, include factors that may change through no fault of your own. These might be environmental conditions such as humidity, processing variables like stirring rate or changes made in the type and amount of ingredients in a formulation. Be realistic when setting the ranges of the factors: You might be surprised at what can happen to your system or what your customers might do with it. W. Edwards Deming provided good advice on factor-setting – he said "start with strata near the extremes of the spectrum of possible disparity…as judged by the expert in subject matter." In the case of the bread-maker, processing factors could not be changed easily due to the nature of the machine, and controls for variables such as ambient temperature were not available in the home kitchen. However, it was not only easy, but logical to substitute various ingredients (cheaper ones!) for the bread mix and hope that these would have no effect.

If you convince yourself that your system is rugged and simply want to prove this to authorities such as the FDA, choose a minimum-run, resolution III design such as a saturated fractional two-level factorial, for example 7 factors in 8 runs. On the other hand, if you suspect that the system may fail due to some untested combination of factors, choose a resolution IV screening design. This medium-resolution choice still allows you to study many factors in few runs. For example, you can study up to 8 factors in only 16 runs. But unlike the low resolution saturated designs, the resolution IV designs give relatively clear estimates of main effects and some information on the presence of two-factor interactions.

What you do as a result of running a resolution IV design depends on which, if any, effects come out significant. Here's a general strategy for follow-up:

- Scenario 1,nothing significant: Remain vigilant for other factors that may affect your response(s).
- Scenario 2, only main effects significant: Change factors to their best levels or search for ways to make the system robust* to these factors.
- Scenario 3, two-factor interaction(s) significant: De-alias the interactions by performing a semifold. Then change factors to their best levels or search for ways to make the system robust to

these factors. For example, knowing that bread-making can fail if a consumer combines water with regular flour, the manufacturer of the machine could try modifying their pre-set processing to be make this combination work. Possibly with a bit more temperature and/or time and/or mixing the bread would rise regardless.

By following this general strategy you will increase your odds of uncovering previously unknown main effects and interactions at a relatively minimal cost in experimental runs. This is an ideal situation—akin to baking your bread and eating it too. Consider applying ruggedness testing in this manner to your manufacturing processes, products and systems of any kind that must be released for use by others.

References

(1) W. J. Youden and E. H. Steiner, *Statistical Manual of the AOAC*, Association of Official Analytical Chemists, Washington D.C., 1975.

(2) Mark J. Anderson and Paul J. Anderson, "Design of Experiments for Process Validation," *Medical Device and Diagnostic Industry*, January 1999, pp. 193-199.

(3) Mark J. Anderson and Patrick J. Whitcomb, *DOE Simplified, Practical Tools for Experimentation*, Productivity, Inc., New York, 2000, Chapter 6 "Getting the Most from Minimal-Run Designs."

(4) Design-Expert® software, Version 6, Stat-Ease, Inc., Minneapolis (http://www.statease.com), 2002.

(5) Patrick J. Whitcomb and Kinley Larntz, *The Role of Pure Error on Normal Probability Plots*, Transactions of Annual Quality Congress of the American Society of Quality (Milwaukee, WI.), 1992.

(6) Douglas C. Montgomery, *Design and Analysis of Experiments, 5th Edition*, John Wiley and Sons, New York, 2001, Appendix XIII.

(7) R. W. Mee and M. Peralta, "Semifolding 2^{k-p} Designs," *Technometrics*, Vol. 42, No.2 (May 2000), pp. 122-143.

(8) Mark J. Anderson and Patrick J. Whitcomb, *How To Save Runs, Yet Reveal Breakthrough Interactions, By Doing Only A Semifoldover On Medium-Resolution Screening Designs*, Transactions of Annual Quality Congress of the American Society of Quality (Milwaukee, WI.), 2001.

Acknowledgments

I wish to thank Patrick J. Whitcomb, my partner at Stat-Ease, for inspiring me to experiment with the DOE methods detailed in this article.

Mark J. Anderson is a principal at Stat-Ease, Inc., Minneapolis. He earned B.S.Ch.E. and M.B.A. degrees at the University of Minnesota. His e-mail address is *Mark@StatEase.com*.

*Here's a 'sidebar' on the terminology of "Ruggedness" versus "Robustness" that I uncovered while doing the research for my bread DOE. Ruggedness is a term that's been used for several decades, primarily for application to analytical method development, for testing possible sources of variation. I discovered that more recent (within the last decade) handbooks and articles on assay validation assign the term "ruggedness" to "external" conditions such as when and where the assay is done, ambient temperature and humidity, alternative sources of raw materials, and lot-by-lot changes; in other words, variables that normally cannot be controlled. On the other hand, the term "robustness" now seems to be in favor for those variables that are "internal" to the assay (or process or product); that is, variables that normally can be controlled. For example, the USP (U.S. Pharmacopeia) and ICH (International Conference on Harmonization of Technical Requirements for Registration of Pharmaceuticals for Human Use) define robustness as "a measure of its [an analytical method] capacity to remain unaffected by small but deliberate variations in method parameters." Doing this, typically with the aid of DOE, "provides an indication of its reliability during normal use." Method parameters include changes in input factors, such as processing time and temperature, composition of reagents, etc. Both types of variables, "rugged" versus "robust," must be included in a proper validation test, so it really becomes just a matter of semantics.