An Introduction to Experimental Design Using SAS®

Jonas V. Bilenas, JP Morgan Chase, Wilmington, DE

ABSTRACT

We will look at how to design experiments using SAS/QC. Examples will come from experiments testing consumer preference for direct mail credit card offers. Discussion will focus on designs of experiments using PROC OPTEX, sample size requirements, and response surface models using PROC LOGISTIC and PROC GENMOD. We will discuss what to do with design results, focusing on simulation and optimization.

INTRODUCTION

Experimental Design is the well defined plan for data collection, analysis and interpretation. The process will help answer your questions about hypotheses you have about how different factors influence a response (or dependent) variable. We often begin the design session by asking the question, “What do you want to know?”

In this paper, we will review the following topics:

- Single Factor: “Test VS Control” Designs
- Design of Multi-Factor Tests
- Sample Size Requirements
- Response Surface Plots
- Extending results to Profitability

Applications will come from examples in direct mail solicitations of credit card offers. The hypothetical design used in this paper try to answers questions about consumer response sensitivities to various components of credit card offers.

SINGLE FACTOR: “TEST VS. CONTROL” DESIGN

Historical use of experimental design in the credit card industry has been the analysis of single factor designs with 2 levels of variation. The advantages of the design is that the test is easy to implement and easy to evaluate. The disadvantages are that the design provides to narrow a view of the universe. Test vs Control designs offer no understanding of interactions among factors. These designs are often wasteful of resources and provide little understanding of what effects the desired response. Some of these designs, also called “Champion vs. Challenger” often vary more than one factor making it impossible to tell what factor resulted in the change in response.

An excellent quote from J. Stuart Hunter on one factor tests:

> “The statistical design of experiments had its origins in the work of Sir Ronald Fisher… Fisher showed that, by combining the settings of several factors simultaneously in special arrays (experimental designs), it was possible to glean information on the separate effects of the several factors. Experiments in which one factor at a time was varied were shown to be wasteful and misleading.”

DESIGN OF MULTI-FACTOR TESTS

The design methodology we will be focusing on is called “Response Surface Methods”. The methodology has been successfully used since the mid 1950’s in various disciplines including engineering, physics and psychology. Historical references are included at the end of the paper.

Response surface methods are designed to evaluate how certain FACTORS (or independent variables) affect a RESPONSE (dependent variable) outcome. The design factors are left in original numerical scale so that the statistically significant functional relationship, derived from a regression model, between factors and response can used to solve for local minimums or maximums. Numerical scales are classed as follows:

- NOMINAL SCALE: Categorical Variables: Expressed in terms of dummy (binary) variables. Used in CLASS statements.
- ORDINAL SCALE: Numerical Values represent relative rank orderings but differences cannot be evaluated. Example: ranking 4 beers from 1 to 4 with 4 being the best. The difference between beers ranked 4 and 3 may not be the same as the difference between 3 and 2.
- INTERVAL SCALE: Ordered with a constant scale, but no true 0. Ratios have no meaning. An example is temperature.
- RATIO SCALE: Ordered with a constant scale and true 0. Examples are weight, age, currency, or duration of low introductory rate.

In this paper, we will use the term “response” to refer to any “dependent variable” that we wish to predict. It some examples, it will be a response rate, but it can be any variable type (nominal, ordinal, interval or ratio).

EXAMPLE OF MULTI-FACTOR DESIGN

We are interested in learning more about consumer price sensitivity to various components of a credit card offer for new accounts. The factors we are interested in exploring are:

- Introductory Rate (**INTRO**) that is applied to Balance Transfers.
- **DURATION**: How long the INTRO rate is good for.
- **GOTO** Rate: APR after INTRO period expires and the rate for purchases.
- Annual **FEE**
- **COLOR** of Envelope. An advertising agency wants to test if a RED envelope will increase response over a White envelope.

In order to measure the effects of the above factors on response rate we need to add variability to each of the factors. Here are the dimensions of the factor levels for each factor that we wish to explore:

INTRO	DURATION	GOTO	FEE	COLOR
0%	6 months	Prime+3.99	$0	RED
1.99%	9 months	Prime+4.99	$15	WHITE
2.99%	12 months	Prime+5.99	$45	

If we look at the number of possible design points from the above table we see that there are 162 possible combinations of offers we can test.

3*3*3*3*2 = 162

This maybe to prohibitive if we wish to minimize test mailings to a small number of mailed offers and mail quantity. We can reduce the number of required points if we focus on the functional form of the model we will build after test results are received. If we limit the model to all main effects, all 2-way interactions, and square terms for non-nominal variables, we can determine how many regression parameters we will need to solve. This is shown on the next page.

Regression Effect	Number of Coefficients
Intercept	1
Main Effects	5
2-way Interactions	10
Square Terms	4
TOTAL	20

In order to measure experimental error we will also need to add 10 additional design points. The total design space can now be limited to 30 points. So we now need to come up with 30 out of 162 possible design points. Wow, how many ways of choosing 30 out of 162? How do we know one design is better than another? We need an optimization method to pick better designs.

WE NEED PROC OPTEX

SAS/QC has a number of experimental design building tools. PROC OPTEX will optimize designs based on the factors specified, levels of factors and model statement. We have specified all of this, so let's see how to use PROC OPTEX.

I often smile when I am asked, "What are the data requirements for experimental design?" The requirements are, "What do you want to know?" Sitting with the user and defining his/her goals, factors, factor levels and design constraints are the major requirements to design an optimal design.

To make PROC OPTEX even simpler, let's set up some user defined inputs. Set-up code:

```
****************************** set up **************************;
%let title   = NESUG 2005 TEST DESIGN;
%let var     = intro duration goto fee color;                    ❶
%let class = color                                               ❷
%let factors = intro=3 duration=3 goto=3 fee=3 color=2;          ❸
%let levels  =  intro     nvals=(0 1.99 2.99)                    ❹
                duration  nvals=(6 9 12)
                goto      nvals=(3.99 4.99 5.99)
                fee       nvals=(0 15 45)
                color     cvals=('RED' 'WHITE')
                ;
%let model   = intro|duration|goto|fee|color@2                   ❺
               intro*intro duration*duration goto*goto fee*fee
               ;
****************************************************************;
```

Some comments about above set up code:

1. We list all variables used in the model in line ❶.
2. We list out any CLASS variables (nominal variables) in line ❷. Make sure to add this variable in line ❶.
3. Line ❸ lists all variables and specifies how many levels are required for each factor.
4. Line ❹ lists levels for each factor. Numeric factors get listed with a NVALS statement and character variables get a CVALS specification.

Let's continue with the code. The rest of the code is driven off of the set-up section. First, let SAS generate the design data set with all 162 possible combinations of offers. This can be done with a DATA step or using PROC PLAN.

```
PROC PLAN ORDERED seed=940522;
  FACTORS &factors
          /NOPRINT;
  OUTPUT OUT=ENUM
         &levels
  ;
Run;
```

LOG Output confirms the 162 design points:

```
NOTE: The data set WORK.ENUM has 162 observations and 5 variables.
```

The DATA created form the run is called ENUM. We can modify this data set if there are prohibitive design points in the model. Evaluate each prohibitive point constraint. Don't eliminate points just because the user feels that certain combinations will never be rolled out. This is just a test to measure relationships between factors and response.

Now we get to the PROC OPTEX which will select design points based on an optimization strategy. In this code we selected what is termed D-Optimality. The procedure iterates over many point combinations looking to maximize the determinant of the matrix (X'X).

```
PROC OPTEX DATA=ENUM SEED=112358;
  CLASS &class;
  MODEL &model;
  GENERATE ITER=1000
           criterion=D
           ;
  OUTPUT OUT=DSGN1;
  title &title;
run;
```

The ITER=1000 option is probably over-kill, but the run on a PC is fast. Selected output is shown here. We are requesting to output the best design which is numbered 1.

```
     The OPTEX Procedure

                                                                   Average
                                                                 Prediction
      Design                                                      Standard
      Number     D-Efficiency     A-Efficiency     G-Efficiency     Error
      ─────────────────────────────────────────────────────────────────────
           1          53.1822          25.8988          82.6242      0.8372
           2          53.1668          25.6013          82.8199      0.8377
           3          53.1400          22.9415          79.3966      0.8554
           4          53.0766          25.7855          79.5299      0.8370
           5          53.0461          26.2382          80.8134      0.8363
           6          53.0376          24.6737          81.2857      0.8471
           7          53.0229          25.9422          81.1176      0.8368
           8          53.0165          25.6451          80.7522      0.8372
           9          53.0018          25.6496          79.7641      0.8402
          10          52.9932          26.3599          81.9653      0.8346
      rows 11-999 were omitted
        1000         50.1626          21.8998          74.4684      0.8934
```

We can now print out the design points:

```
proc sort data=dsgn1;
  by &var;

PROC PRINT DATA=DSGN1;
run;
```

Output:

```
      Obs    intro    duration    goto    fee    color
        1     0.00        6       3.99      0    WHITE
        2     0.00        6       3.99     45    RED
        3     0.00        6       5.99      0    RED
        4     0.00        6       5.99     45    WHITE
        5     0.00        9       3.99     45    RED
        6     0.00        9       4.99     15    WHITE
        7     0.00        9       5.99      0    RED
        8     0.00       12       3.99      0    RED
        9     0.00       12       3.99     45    WHITE
       10     0.00       12       5.99      0    WHITE
       11     0.00       12       5.99     45    RED
       12     0.00       12       5.99     45    WHITE
       13     1.99        6       3.99     15    RED
       14     1.99        6       4.99     45    WHITE
       15     1.99        6       5.99      0    WHITE
       16     1.99        9       5.99     45    RED
       17     1.99       12       3.99      0    WHITE
       18     1.99       12       5.99     15    RED
       19     2.99        6       3.99      0    WHITE
       20     2.99        6       3.99     45    WHITE
       21     2.99        6       4.99      0    RED
       22     2.99        6       5.99     15    WHITE
       23     2.99        6       5.99     45    RED
       24     2.99        9       3.99      0    RED
       25     2.99       12       3.99     15    WHITE
       26     2.99       12       3.99     45    RED
       27     2.99       12       4.99      0    WHITE
       28     2.99       12       4.99     45    RED
       29     2.99       12       5.99      0    RED
       30     2.99       12       5.99     45    WHITE
```

Let’s see a design point summary:

```
      proc tabulate
        data=dsgn1 noseps;
        class &var;
        table &var all

              ,
                 n='Design Points'
                *f=comma15.
                 /rts=10;
      run;
```

```
┌─────────┬──────────────┐
│         │ Design Points│
├─────────┼──────────────┤
│intro    │              │
│0        │            12│
│1.99     │             6│
│2.99     │            12│
│duration │              │
│6        │            12│
│9        │             5│
│12       │            13│
│goto     │              │
│3.99     │            12│
│4.99     │             5│
│5.99     │            13│
│fee      │              │
│0        │            12│
│15       │             5│
│45       │            13│
│color    │              │
│RED      │            15│
│WHITE    │            15│
│All      │            30│
└─────────┴──────────────┘
```

Note that center points get fewer design points than extremes. Since the confidence intervals are widest at the extremes of the design more points are added at extremes to minimize the confidence interval of estimation the further one gets away from the center. Also note that if no square terms are added to the model statement then no center points are included in the design.

SAMPLE SIZE REQUIREMENTS

If this was an engineering design, the sample size is the 30 test points. In this case, we wish to look at response rate to these offers and build a logistic regression model after test results arrive. We have to estimate mailbase required for each design point.

We can use some of the sample size estimation procedures used for testing proportions in Test vs. Control designs. Let us use PROC power to determine sample size requirements for 2 proportions with expected response rate of 0.4% and 0.6%, 2-tail test, a 95% confidence interval, and a 50% power.

```
proc power;
   twosamplefreq test=fisher
     groupproportions = (.004 .006)
     npergroup = .
     power = .50
     sides = 2
     alpha=.05;
run;
```

Resulting Output:

```
The POWER Procedure
Fisher's Exact Conditional Test for Two Proportions

             Fixed Scenario Elements

Distribution                     Exact conditional
Method                Walters normal approximation
Number of Sides                                  2
Alpha                                         0.05
Group 1 Proportion                           0.004
Group 2 Proportion                           0.006
Nominal Power                                  0.5

Computed N Per Group

Actual     N Per
 Power     Group

 0.500     10444
```

There currently is no power calculation for LOGISTIC models in SAS9. However, we may want to determine mail quantity by running a PROC POWER for REGRESSION since we may want to evaluate a ratio response such as NPV or total balance transfer for booked accounts. We would take the results of the following PROC POWER and divide by the total net response rate for another estimate of mail quantity per design point.

```
proc power;
  multreg
  model = random
  nfullpredictors = 19
  ntestpredictors = 1
  partialcorr = 0.35
  ntotal = .
  power = .5;
 run;
```

Output:

```
Computed N Total

Actual        N
 Power    Total

 0.502       49
```

Using 50 as the required number of accounts, and an expected overall gross response rate of 0.50% and approval rate of 90%, the total number of mail pieces per cell works out to:

50/(0.0050 * 0.90) = 11,1111

For this hypothetical design, let us use 12,000 pieces mailed for each of the 30 design points. The next section looks at hypothetical results.

RESPONSE SURFACE PLOTS

Results come in for the mailing and we decide to model response rate using PROC LOGISTIC using the model specified in the design set up. We could also have used PROC GENMOD.

Results of TEST:

Design Point	intro	duration	goto	fee	color	quantity	Respond
1	0	6	3.99	0	WHITE	12,000	22
2	0	6	3.99	45	RED	12,000	18
3	0	6	5.99	0	RED	12,000	24
4	0	6	5.99	45	WHITE	12,000	36
5	0	9	3.99	45	RED	12,000	48
6	0	9	4.99	15	WHITE	12,000	108
7	0	9	5.99	0	RED	12,000	114
8	0	12	3.99	0	RED	12,000	174
9	0	12	3.99	45	WHITE	12,000	72
10	0	12	5.99	0	WHITE	12,000	144
11	0	12	5.99	45	RED	12,000	70
12	0	12	5.99	45	WHITE	12,000	67
13	1.99	6	3.99	15	RED	12,000	60
14	1.99	6	4.99	45	WHITE	12,000	24
15	1.99	6	5.99	0	WHITE	12,000	21
16	1.99	9	5.99	45	RED	12,000	25
17	1.99	12	3.99	0	WHITE	12,000	84
18	1.99	12	5.99	15	RED	12,000	72
19	2.99	6	3.99	0	WHITE	12,000	28
20	2.99	6	3.99	45	WHITE	12,000	18
21	2.99	6	4.99	0	RED	12,000	16
22	2.99	6	5.99	15	WHITE	12,000	21
23	2.99	6	5.99	45	RED	12,000	18
24	2.99	9	3.99	0	RED	12,000	22
25	2.99	12	3.99	15	WHITE	12,000	36
26	2.99	12	3.99	45	RED	12,000	43
27	2.99	12	4.99	0	WHITE	12,000	72
28	2.99	12	4.99	45	RED	12,000	42
29	2.99	12	5.99	0	RED	12,000	60
30	2.99	12	5.99	45	WHITE	12,000	36

PROC LOGISTIC CODE:

```
proc logistic data=test descending;
  class color;
  model respond/quantity =
               intro|duration|goto|fee|color@2
               intro*intro duration*duration goto*goto fee*fee
               /sle=.05 sls=.05 selection=stepwise;
run;
```

Selected output from the LOGISTIC run:

```
      Analysis of Maximum Likelihood Estimates


                                                  Standard           Wald
      Parameter              DF      Estimate        Error     Chi-Square    Pr > ChiSq
      Intercept               1       -8.8506       0.6529       183.7708        <.0001
      intro                   1       -0.0252       0.0875         0.0829        0.7734
      duration                1        0.6275       0.1444        18.8814        <.0001
      intro*duration          1       -0.0287      0.00808        12.6353        0.0004
      fee                     1        0.0308      0.00901        11.6605        0.0006
      intro*fee               1       0.00283     0.000952         8.8321        0.0030
      duration*fee            1      -0.00202     0.000526        14.8228        0.0001
      duration*duration       1       -0.0206      0.00767         7.1792        0.0074
      fee*fee                 1      -0.00057     0.000150        14.4006        0.0001
```

Note that for stepwise model in PROC LOGISTIC a non significant main effect will be retained if the interaction terms involving the main effect are significant. Looking at main effects; high intro reduces response, and high duration increases response. The GOTO effect and COLOR were not a significant factor in predicting response rates. It is hard to tell what is going on with the fee term. To further investigate and to report results, let's set up some surface plots. We will vary 2 terms and hold the third term constant and plot response rate as the Z axis.

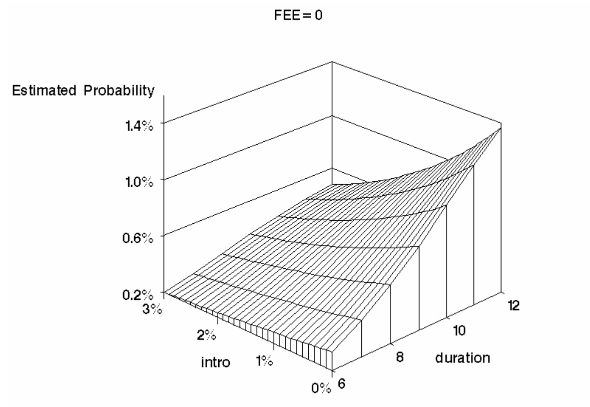
Here is the code that generates surface plot data and the plots:

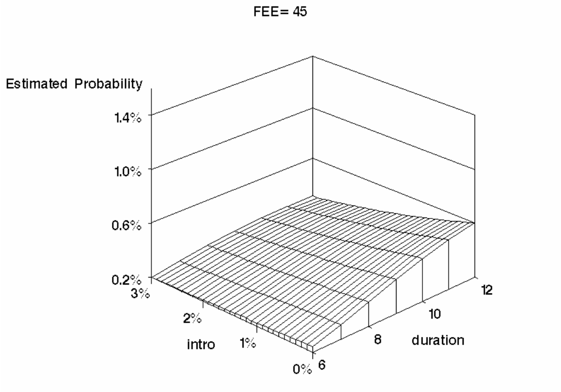
```
      data surface;
        do intro = 0 to 3 by .1;
          do duration = 6 to 12;
             do fee=0 to 45;
               goto=5.49; color="WHITE";
               output;
             end;
         end;
        end;

      proc append base=test data=surface force;

      proc logistic data=test descending;
        class color;
        model respond/quantity =
                     intro|duration|goto|fee|color@2
                     intro*intro duration*duration goto*goto fee*fee
                     /sle=.05 sls=.05 selection=stepwise;
        output out=score p=p;
      run;
```

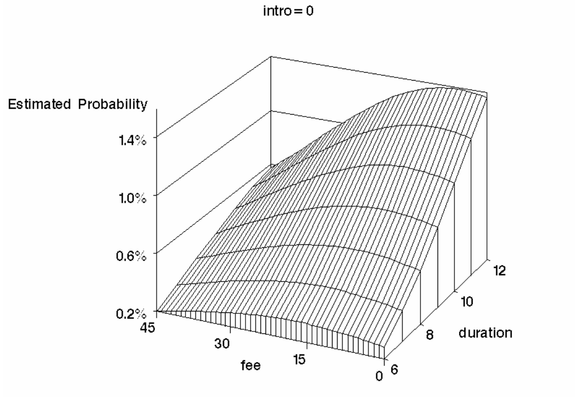
```
proc format;
  picture pct (round) low-high ='9%';
proc g3d data=score;
  where respond=. & fee=0;
  plot intro*duration=p/grid  side zmax=0.014 rotate=45 tilt=65;
  format p percent6.1 intro pct.;
  title FEE=0;
run;
proc g3d data=score;
  where respond=. & fee=45;
  plot intro*duration=p/grid  side zmax=0.014 rotate=45 tilt=65;
  format p percent6.1 intro pct.;
  title FEE=45;
run;
proc g3d data=score;
  where respond=. & intro=0;
  plot fee*duration=p/grid  side zmax=0.014 rotate=65 tilt=65;
  format p percent6.1 intro pct.;
  title intro=0;
run;
proc g3d data=score;
  where respond=. & intro>-2.99;
  plot fee*duration=p/grid  side zmax=0.014 rotate=65 tilt=65;
  format p percent6.1 intro pct.;
  title intro=2.99;
run;
```

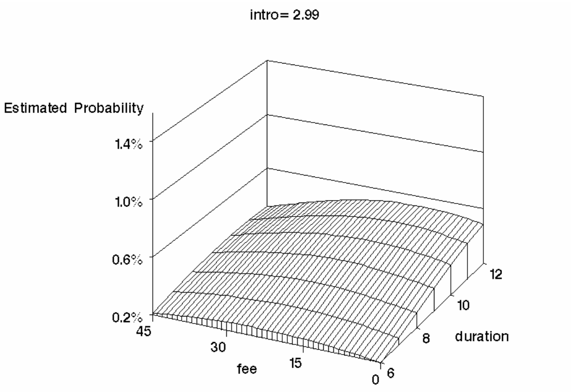
Response surface plots are shown here. These were merely cut and pasted from the SAS output window. We could use PROC GREPLAY to present 4 plots on a page as well.

Response Surface Plots









EXTENDING RESULTS TO PROFITABILITY

Response rates make sense. Consumers want the best offer for little or no price. The question that we have to ask is if we can make money as a corporation by making such offers? We can extend the surface plot to a NPV plot. This may not be trivial since it requires models for time series components of NPV. Generally the models are also of a regression form, using factors and time components to predict performance of accounts. With an NPV estimation derived from experimental design data we can utilize PROC LP to make optimal decisions on what price to offer different populations. These complexities are left to the reader and/or a subsequent paper.

ADDITIONAL CONSIDERATIONS:

- **Include a Replicate Point.** This will test if randomization was done correctly. If the replicate point does statistically different response results one would question whether randomization was done correctly. If your mail processing shop uses ORACLE instead of SAS or DB2 keep in mind that ORACLE does not have randomization functions.
- **Removal of Design Points.** One can remove design points from the design if they are not feasible. This should be done after PROC PLAN and before PROC OPTEX. However, don't be too quick to remove points. Remember that this is a test and not a roll-out. Do not leave this procedure to the business user since removal of design points may invalidate the test by causing collinear relationships among factors.

- **How to Treat Segmentation covariates.** There are a number of ways of treating segmentation or population groups:
    i. Treat as a factor in the design.
    ii. Model as a covariate in the analysis
    iii. Replicate design in all segments.
- **Do not add confounding conditions.** For example, mailing better responders using 1st class and mailing lower responding names using 3rd class. If your “Business as usual” decision is to cut mail base by response score cut-offs, then use the same cut-off strategy for each design point.

CONCLUSION

Experimental Design is simple with the power of SAS. You have to ask; “What do you want to know?”, “What do you want to investigate?”, and “What are your constraints in terms of sample size and design points?” Run a response surface analysis using PROC LOGISTIC or PROC GENMOD or PROC GLM. Consider using the experimental design to feed the data for a profitability simulation and optimization.

REFRENCES:

- Cragle, R.G., R.M. Meyers. R.K Waugh, J.S.Hunter, & R.L. Anderson (1955). The effects of various levels of sodium citrate, glycerol, and equilibration time on survival of bovine spermatoza after storage at -790C, J. Diary Science, 38, 508.
- J. Stuart Hunter (1987). Applying Statistics to Solving Chemical Problems. Chemtech, 17, 167.
- Myer, D.L. (1963). Response surface methodology in education and psychology, J Exp. Educ., 31, 329.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jonas V. Bilenas
JP Morgan Chase Bank
Mail Code DE1-1272
One Christina Center
301 N. Walnut Street, 15th Floor
Wilmington, DE 19801
302-282-2462
Email: Jonas.Bilenas@chase.com
jonas@jonasbilenas.com
Web: http://www.jonasbilenas.com

This work is an independent effort and does not necessarily represent the practices followed at JP Morgan Chase Bank.