Categorical Data in a Designed Experiment Part 4: Estimating Power of Test Designs for a Binary Response

Authored by:

Sarah Burke, PhD
Melissa Key, PhD
Krista Wurscher

17 August 2021

The goal of the STAT COE is to assist in developing rigorous, defensible test strategies to more effectively quantify and characterize system performance and provide information that reduces risk. COE products are available at https://www.afit.edu/STAT.

STAT Center of Excellence
2950 Hobson Way – Wright-Patterson AFB, OH 45433

Table of Contents

Executive Summary

How should you evaluate a test design if your response is *not* continuous and is, in fact, binary? For example, suppose your system is a one-use system that has a reliability requirement specified as a proportion which must be assessed across various conditions. Alternatively, suppose you have a communication system under development with a response of whether or not a "message successfully delivered," which is measured with a pass/fail metric. How can we assess the effectiveness of a proposed test design for these types of responses? This best practice describes a technique to estimate the power of a test design for a binary response using Monte Carlo simulation and describes how to use an online application developed by the Scientific Test and Analysis Techniques Center of Excellence (STAT COE) to implement the method.

Keywords: design evaluation, Monte Carlo simulation, power analysis

Introduction

The Scientific Test and Analysis Techniques Center of Excellence (STAT COE) has long emphasized the need to evaluate a test design prior to test execution to understand its risks and assess its effectiveness at accomplishing the test objective. Some methods to evaluate a test design include graphs to assess test space coverage, design aliasing, prediction variance, and estimated power (Harman, 2018). If the response in the experiment is continuous and follows an (approximate) normal distribution, evaluating the design is relatively straightforward with the right tools. For example, the statistical software package JMP readily provides all of the previous design evaluation metrics (and more) in their design evaluation report.

In a perfect world, we would evaluate test designs for binary responses using the same methods used for continuous responses. In practice, however, methods to evaluate test designs for binary responses require more assumptions and are less readily available. Consequently, we first encourage you to re-assess the response to determine if there is a way to measure the output as a continuous quantity. Part 1 of this best practice series (Ortiz, 2018a) explains the advantages of using continuous responses; the most prevalent being that continuous responses typically save test resources. However, suppose you cannot convert a binary response to a continuous response. Familiar design evaluation techniques focused on the factor space (e.g., factor aliasing and design coverage) are still applicable for binary responses and can be used as-is. Power analysis, however, cannot be as easily applied.

Part 2 of this best practice series (Ortiz, 2018b) describes techniques to estimate power by adjusting the signal-to-noise ratio. While these approaches are more straightforward to implement, they rely on applying a normal approximation to the binomial distribution and typically assume the test design is a factorial or fractional factorial design. This best practice describes an alternative technique to estimate the power of a test design for a binary response that does not have these limitations. The STAT COE created an online tool (LINK) for practitioners to implement this technique, which we describe in detail later.

Background: Power

Statistical power is the probability of rejecting the null hypothesis when the null hypothesis is false (i.e., the probability of a true positive). In a design of experiments (DOE) context, power can be interpreted as the probability of detecting if a factor influences the response when it really does (given a specified signal-to-noise ratio [SNR]). Power is a function of the significance level (denoted as α, also called the type I error rate), the SNR, the test design itself, and the number of terms in the regression model. Using input values of type I error, SNR, and the desired model, software such as JMP can estimate the power of a design, under the assumption that the response is normally distributed. In this case, there is a closed-form (i.e., known) solution to estimate power based on the F-distribution (Montgomery, 2017). The assumption of the response distribution is critical to correctly interpret the risk associated with the design. For example, if the power for the main effects of a design is estimated as 0.8, then there is a 20% chance that the factor effect on the response will not be detected in the resulting analysis if that factor really does have an effect on the response. Understanding this risk is important as it helps justify the choice and size of a design matrix used in a test.

Challenges with Binary Responses

Binary responses present many challenges in test and evaluation (T&E). They provide less information compared to continuous responses and typically require more test resources to adequately characterize (Ortiz, 2018a) as a function of the factors. In addition, a design created for a continuous response is typically not a good design for a binary response (Johnson and Montgomery, 2009). Nevertheless, binary responses are sometimes unavoidable and the goal of developing an efficient, yet effective test design is still important.

We typically use logistic regression to model the probability of a binary response over a factor space, shown in Equation 1:

$$p(\boldsymbol{x}) = \frac{\exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)}, \quad (1)$$

where $p(\boldsymbol{x})$ is the probability of success, β_i represent the model parameters for $i = 1, \ldots, k$ factors, and x_i represent the factors. This model can be readily expanded to include two-factor interactions or quadratic terms. Unlike traditional linear regression, logistic regression is a nonlinear model, meaning that the model is nonlinear in the parameters – note the "exp" term in both the numerator and denominator of Equation 1.

We could execute a classical DOE, such as a factorial design (Natoli and Oimoen, 2019), to estimate the logistic regression model parameters (β_i in Equation 1). This type of design will achieve maximum power (for a fixed sample size) for a continuous response because the test runs are at the corners of the design space. However, this design is not typically sufficient for binary responses, where we need a mixture of successes and failures to estimate the logistic regression model. Evaluating a design for a binary response using the metrics for a continuous response, therefore, is misleading and does not correctly

estimate the risk associated with the design. Unfortunately, because logistic regression is a nonlinear model, there is no closed-form solution (i.e., fixed formula) to estimate the power prior to test execution. One way to estimate power is using Monte Carlo (MC) simulation.

Simulated Power

We can estimate the power of a proposed test design by simulating results of the DOE thousands of times. A description of the simulation procedure is shown in Table 1.

Table 1: Description of Empirical Power Simulation

Step	*Description*
0	User provides test design, specifies the form of the logistic regression model, initial guesses for model parameters, and the significance level (α).
1	Randomly generate binary responses for all test runs in design using the specified model in step 0.
2	Fit a logistic regression model using randomly generated responses.
3	Determine which model terms are statistically significant by comparing each p-value to the stated significance level.
4	Repeat Steps 1-3 N times (e.g., N = 1000)
5	Calculate estimated power as the proportion of simulations where model term was deemed statistically significant over total N iterations.

Step 5 of the simulation provides the primary desired output. In each iteration of the MC simulation, we make a determination on whether a model term is statistically significant. Power for each model term can be estimated as the proportion of simulations where the term was identified as statistically significant across all iterations of the simulation. This type of power estimate is called an "empirical" power estimate because it is determined using simulation. The primary goal is to assess the effectiveness of the design prior to test execution using subject matter expertise in the simulation.

The factor coefficients (β_i in Equation 1) determine the probability that the simulated response is a success or a failure and are analogous to the effect size (signal) that a model term has on the response. Figure 1 shows the effect of the model parameter in a logistic regression model for a single factor (with $\beta_0 = 1$). Note that the logistic regression model forms an S-shaped curve to model the effect of a single factor on the probability p when β_1 is greater than zero (and a reverse S-shape when β_1 is less than zero). When β_1 is greater than 0, then p increases as x increases. As the magnitude of β_1 increases, the line becomes steeper. When β_1 is 0, then the factor x has no effect on p, and so the line is flat. To estimate power, users must specify estimated values for each model term. Because these values can be challenging to estimate prior to designing the experiment, we have a tool to help users specify these values. Please refer to the Appendix for details on this method.

The rest of this best practice describes how to use the app that implements this technique. We discuss the user inputs and how to interpret all of the outputs. For further questions, email the STAT COE at COE@afit.edu.

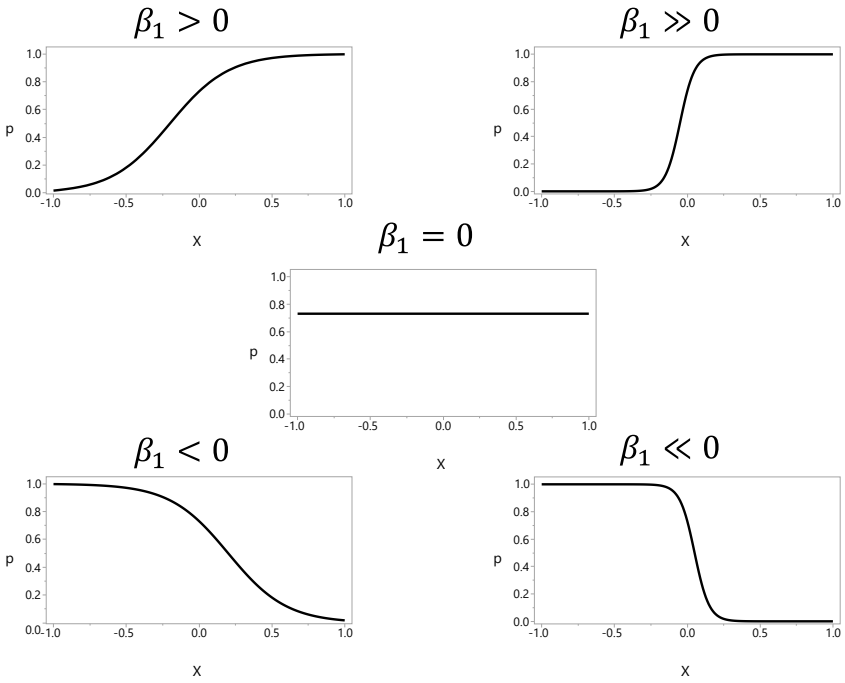


Figure 1: Relationship of logistic regression model coefficients with probability of success

Binary Power Shiny App

The shiny app is available on the STAT COE website at [INSERT LINK]. As an illustrative example, suppose we are planning a test to estimate the probability of penetrating armor. There are three factors of interest: velocity, angle, and lot number. Velocity and angle are both continuous factors while the lot number is categorical with two levels (coded A and B). For the purposes of this example, we leave the factor levels in coded units (-1, 1), where -1 represents the low level and 1 represents the high level for the actual units of the factor. Note that the actual units can be used in the design when using the app. The response is Armor Penetration with a "1" representing that the projectile went through the target and a "0" representing that the projectile did not go through the target.

Initial User Inputs

This section describes the user-required inputs to the app.

Load a Design File

The first step is to load the test design under consideration by clicking the "Browse" button on the main page. The design file should consist of comma, semicolon, or tab delimited columns, with one column per factor. Empty columns (e.g., placeholders for the response variable) are ignored. By default, the app assumes that the first row of each design file contains factor names. If factor names are not included,

deselect the checkbox next to “Does the file have a header row?” and factor names will be generated automatically. Figure 2 shows the initial view of the app before a design file is uploaded.

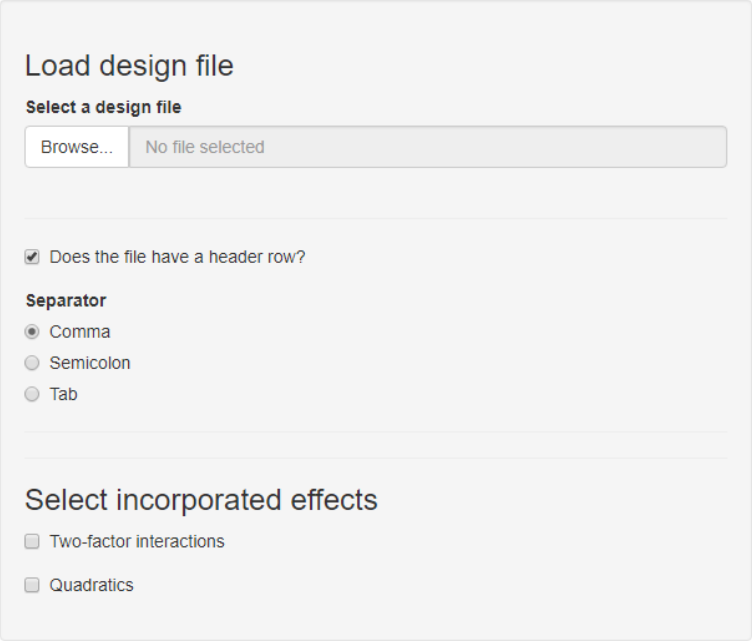


Figure 2: Step 1 to Use Binary Power App - Load the Design File

For the armor penetration example, we created an initial design with 20 runs: a replicated 2^3 design with 4 center points. This design is a logical design choice for continuous responses; however, will it be an effective design for a binary response?

Review Factor Properties

Once the design has been uploaded to the app, review the factor properties to ensure it has been read-in correctly. For each factor identified in the data table, you can specify whether the factor is numeric, categorical, or should be ignored in the analysis. This last option is helpful if there were other columns (e.g., run order) in the design file that are not of interest in the design evaluation. The “Ignore” feature will exclude that column in most subsequent plots, tables, and analyses.

The app provides a summary for each factor. For categorical factors, the frequency and percentage of runs at each level is given. For numeric factors, the summary statistics give the minimum and maximum, along with the number of distinct levels the factor takes on. Figure 3 shows the factor properties for the armor penetration design.

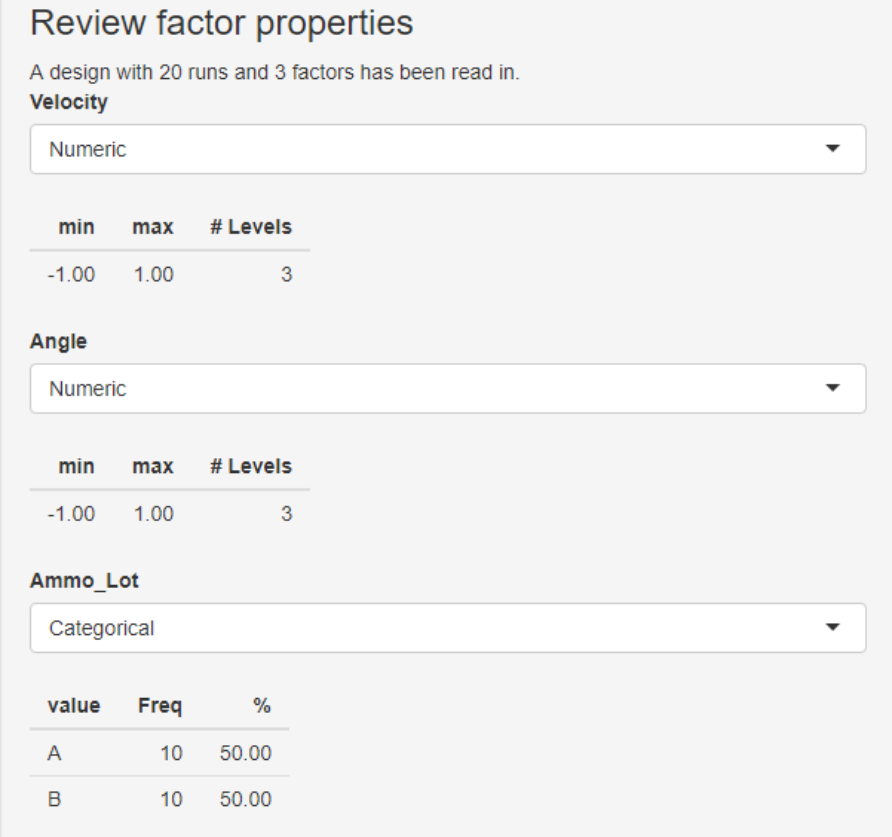


Figure 3: Step 2 to Use Binary Power App – Review Factor Properties

Specify the Form of the Model

Once the factors have been identified, specify the desired form of the model by selecting whether to include two-factor interactions and/or quadratic terms. By default, just main effects are included. Note that while the option to include quadratic effects is always listed, quadratic effects are only possible for numeric factors. When all factors are categorical, clicking on "Quadratics" will bring up an empty table. Figure 4 shows the model specified for the armor penetration example; in this evaluation, we include just the main effects for an initial screening experiment, and thus leave both boxes unchecked.

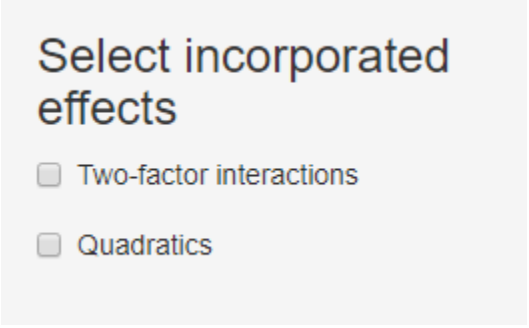


Figure 4: Step 3 to Use Binary Power App - Specify Form of the Model

Specify Model Parameters

Click the second tab in the app, called "Factor Coefficients." The factor coefficients tab contains one to three tables depending on whether main effects, two-factor interactions, and/or quadratic terms are included in the specified model from Step 3 of the app. These tables are critical in the app as they capture the estimated coefficients in the logistic regression model (β_i in Equation 1). These are the values that are used to randomly generate the responses in each iteration of the Monte Carlo simulation. The simulation coefficients determine the probability that the simulated response is a success or a failure described in Table 1, Step 2.

Main Effects Table

The intercept term in the model represents a baseline condition of the factor levels, corresponding to the first level of all categorical factors and all numeric factors set to zero. For numeric main effects, the coefficient is related to the slope of the logistic regression equation (as seen in Figure 1). For categorical main effects, we must specify coefficients for all factor levels except the first (which is incorporated into the baseline condition). For example, if the factor has j = 3 levels, we must specify j - 1 = 2 model coefficients. For the j^{th} level (j > 1), the coefficient is the expected difference (before transforming to a probability in the logit function) between the j^{th} level and the first level when all other factors are fixed at the baseline condition.

Interaction Effects Table

For two numeric factors, the single coefficient is the expected change in slope of one factor as the other factor increases by one unit. For two categorical factors, we must specify coefficients for each combination of non-baseline factor levels. For example, if factor 1 has j = 3 levels and factor 2 has k = 4 levels, we must specify (j - 1)*(k - 1) = 6 coefficients. For factor 1 at level j (j > 1) and factor 2 at level k (k > 1), the coefficient gives the expected difference in the response between that condition and the baseline condition (both factors are set to level j = 1 and k = 1). For one categorical factor and one numeric factor, we must specify a coefficient for each level of the categorical factor except for the baseline condition. In other words, if the factor has j levels, we must specify j - 1 coefficients. The value of the interaction term for the j^{th} level (j > 1) of the categorical factor is the change in the slope of the numeric factor when the categorical factor is set to level j.

Quadratics Effect Table

Quadratic effects only apply to numeric factors. Including quadratic terms in the model allows us to estimate curvature in the response.

Additional Options

The "Estimate" column in each table controls whether or not the term is included in the estimated statistical model (Table 1, Step 3). The coefficients will still be used to simulate the responses; the terms will just not be included in the model evaluation. For categorical factors with more than two levels, despite multiple checkboxes corresponding to each level (or combination of levels), the factor can only be "in" or "out" of the model. To update any element of the factor coefficients table, enter in the appropriate values and then click the "Update Table" button.

Note: the coefficient values can be challenging to estimate without prior knowledge of the system. We recommend using the binary coefficients app, also developed by the STAT COE, available at the following link [INSERT LINK]. This app allows the user to enter in estimates of the probability of success at specified test points. The app then estimates the coefficients of the logistic regression model based on those probability estimates. See the Appendix for more information.

Armor Penetration Example

For the armor penetration test design, we consulted with a subject matter expert to obtain estimates of the probabilities of armor penetration at the corner points of the factor space. The subject matter expert suspected there could be up to a 5% difference between the two ammo lots. See the Appendix for more details on this approach to estimating the model coefficients. Using the binary coefficients app and inputs from the subject matter expertise, we obtained the following estimates for the model coefficients, shown in Figure 5.

Main Effects

Factor	Level	Type	Coefficient	Estimate
(Intercept)		Baseline condtion	-0.27	☑
Ammo_Lot	B	Change from Ammo_Lot = 'A'	0.07	☑
Angle		Slope	-0.35	☑
Velocity		Slope	1.50	☑

Update Table

Figure 5: Specifying Model Coefficients for Armor Penetration Example

Outputs

This section describes the output of the app and how to interpret the results.

Predicted Probability of Success

Once we enter the coefficients into the model, a summary table shows the estimated probability of success for the different factor combinations in the test design. When all the coefficients are set to zero, the probability of success for all factor conditions is 0.5. This table can be useful to assess the validity of the coefficients entered in the previous step.

Assess Design

Click the Assess Design tab. This tab provides two additional design evaluation tools to assess the quality of the test design: 1) a correlation plot to assess the design orthogonality and 2) a scatterplot matrix to assess the test space coverage.

When evaluating a design, we want to make sure that no two effects are highly correlated with each other. If two model terms are highly correlated with each other, we are not able to distinguish which model term causes a change in the response. The correlation plot is a symmetric table with each row and column representing a term in the specified logistic regression model. Black squares represent complete correlation (correlation = 1) between model terms and white squares represent no correlation (correlation = 0) between model terms. On the main diagonal is the correlation between a model effect with itself. Any term is perfectly correlated with itself, so we always see black squares down the main diagonal. In a perfect scenario (called an orthogonal design), we will see no correlation between any other effects in the off diagonal; i.e., white squares entirely in the off-diagonal. Figure 5 shows the correlation plot for the armor penetration example. This was a 2^3 design with 4 center points; as expected, the design is an orthogonal design for the main effects.

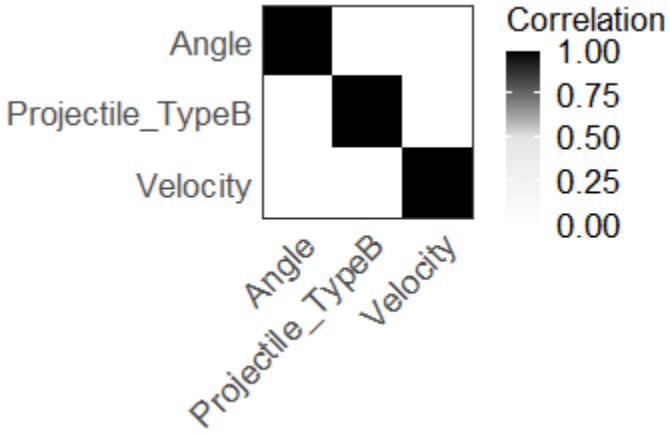


Figure 6: Correlation plot for the Armor Penetration Example

The scatterplot matrix is a set of pairwise scatterplots for each factor in the test design. This plot visualizes the test space coverage to ensure the test space is sufficiently captured in the design. This plot also includes color coding to represent the probability of success in the design space. Red represents high probabilities of success while blue represents low probabilities of success. Note that jittering is included on the plots for categorical factors to distinguish multiple test points at that location. Figure 7 shows the scatterplot matrix for the armor penetration test design example. Note that the predicted probability of penetration is much higher for high velocities, with the highest probability of success at the low angle.

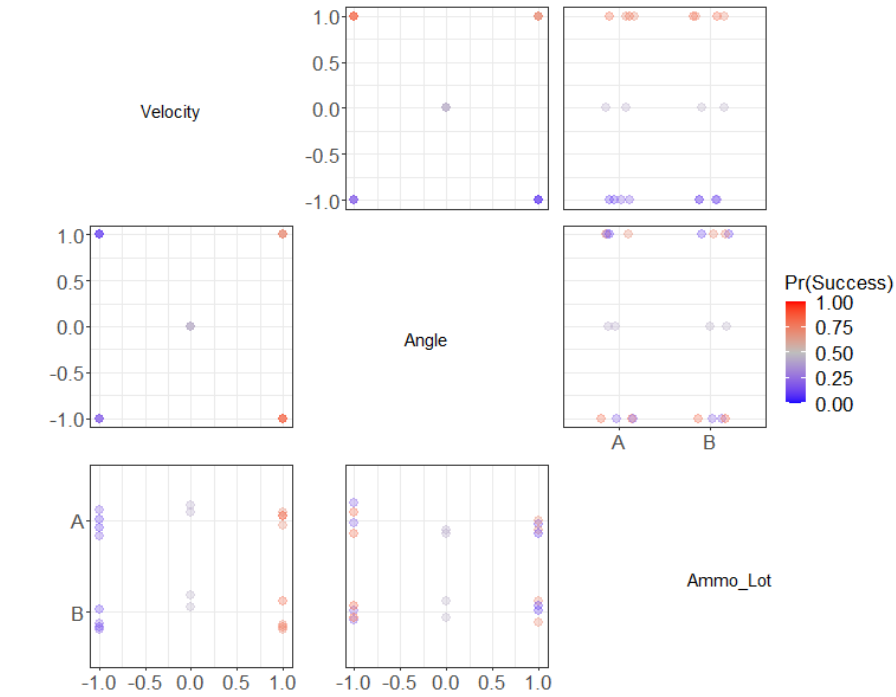


Figure 7: Scatterplot Matrix for Armor Penetration Design

Simulation

Click the Simulation tab; this is where the primary outputs of the app are displayed. Specify the total number of runs in the MC simulation, which has a default value of 1,000 (and a maximum possible value of 10,000). This value determines the number of times to repeat the simulated response values as described in Table 1, Step 4. Click the "Run Simulation" button. The simulation will take several seconds to execute. Note that the results in the simulation will vary if you execute the simulation multiple times due to the random nature of the MC simulation.

The app displays several results from the simulation to assess the effectiveness of the design, which we discuss in the following sub-sections.

Table of Results

The primary table of results summarizes the output of the MC simulation including: the mean model coefficients, standard error of the model coefficients, 95% confidence interval of the coefficient mean estimate, estimated power for various significance levels (α), and a heuristic to assess the likelihood of "separation." We discuss each of these elements in turn. As described in Table 1, Step 2, in each iteration of the MC simulation, the model coefficients are estimated for the randomly generated data in the design. The table provides the mean model coefficient estimate across all N iterations of the MC simulation for each model term. Ideally, the mean model coefficient is close to the user-specified value provided in the "Factor Coefficients" tab.

In addition to the mean coefficient value, the table also displays the standard error for each model coefficient. This value is important to evaluate to ensure the design has a sufficient number of runs to

estimate the logistic regression model. As discussed in Part 3 (Natoli et al., 2020) of this best practice series, logistic regression models are susceptible to separation, a condition when a factor or a model term perfectly predicts the response. When separation occurs, the model terms in the logistic regression model do not exist because an infinite number of solutions could fit the data. Large values of standard error of the model coefficients are an indicator that separation exists in the data, which can be hard to detect through visual inspection as the number of factors increases. Definition of "large" depends on the units of measure of the factors; however, some authors have suggested some heuristics to detect separation. SAS (2019) for example, suggests that standard errors above 5000 may be useful to detect separation. The proportion of MC iterations with standard error over 5000 is shown in the last column of the table. If separation is likely to occur, the chosen design will not be effective because the logistic regression model (or some of the model terms) will not be estimable. Separation is more likely to occur with small test sizes, when there are a large number of model terms relative to the sample size, and there aren't a sufficient number of levels of all factors (Agresti, 2013). In cases where separation is likely to occur, we recommend considering alternate design choices to minimize this issue occurs in the test.

Figure 8 shows results for an MC simulation with N = 1000 iterations for the armor penetration example. The table shows the actual parameter values (those specified in the Factor coefficients tab, Figure 5) and the mean estimated values from the MC simulation. All of the mean estimates are in the correct direction as specified, although the magnitudes vary. Note there is also a metric that states "92 out of 1000 runs produced probabilities of 0 or 1." This indicates that in 92 of the 1000 MC iterations, the response values in the design were either all 0s (no penetration) or all 1s (penetration). Therefore, 9.2% of the time, a logistic regression model is not estimable as the responses were all the same values across all 20 runs.

92 out of 1000 runs produced probabilities of 0 or 1.

Show All entries

	Factor	Actual	Mean Estimate	Std. Error	95% CI for Mean
1	(Intercept)	-0.27	-1.11	6.43	(-13.716, 11.506)
2	Ammo_LotB	0.07	0.184	9.69	(-18.807, 19.174)
3	Angle	-0.35	-1.82	5.39	(-12.375, 8.74)
4	Velocity	1.5	7.81	11.5	(-14.725, 30.354)

Figure 8: Binary Power App Table of Results (Part 1)

The results table then shows the "empirical power" estimates from the MC simulation for different values of type I error (α). As described in Table 1, Step 3, we determine whether a model term is significant in the logistic regression model by comparing the p-value to a specified significance level. If the p-value is less than α, then we deem the term statistically significant; if the p-value is greater than α, the model term is not statistically significant. The estimated power is the proportion of MC iterations where the model term was deemed statistically significant. There are power estimates for four values of

α: 0.01, 0.05, 0.1, and 0.2. These values correspond to confidence levels of 0.99, 0.95, 0.9, and 0.8, respectively. Ideally, these power estimates will be relatively large (≥ 0.80) for model terms with important effects as specified in the Factor Coefficients tab. If the estimated power values are low, then we recommend considering alternative design options (e.g., increase the number of runs or levels of the factors), particularly for factors that likely have important effects on the response.

Figure 9 shows the remainder of the table of results for the armor penetration example. Note that this table has been cropped to show the factor labels with the estimated power values. In this example, the power for all model terms is well below 0.80, even for the largest value of α. For example, the power to detect the effect of velocity (which had the largest effect of the three factors) is estimated to be just 0.572 for a significance level of 0.2. Note that the power for the ammo lot is quite small, which may be satisfactory if the predicted effect of the ammo lot is not considered practically important. Remember that having high power means that we can detect a factor effect with a given SNR. If the expected SNR is small, then low power will be expected for limited test sizes. The user must decide how large an effect needs to have a large probability of detecting. The last column shown in Figure 9 is the separation heuristic previously described. In this example, separation is likely to have occurred at least 10% of the time for the velocity effect.

92 out of 1000 runs pro
Show All entries

	Factor	α = 0.01	α = 0.05	α = 0.1	α = 0.2	% Std. Err > 5000
1	(Intercept)	0	0.004	0.057	0.231	6.7
2	Ammo_LotB	0	0.007	0.05	0.155	8.3
3	Angle	0	0.007	0.017	0.059	8.7
4	Velocity	0	0.334	0.473	0.572	10.1

Showing 1 to 4 of 4 enti Previous 1 Next

Figure 9: Binary Power App Table of Results (Part 2)

Power Plot

This figure provides a visualization of the estimated power shown in the table of results. The graph displays a power curve against potential values of the significance level; there is a curve for each model term. Ideally, we will see high values of the estimated power for all the model terms; a reference line for 0.8, a common standard for power, is included in the plot. This figure can provide insight into the tradeoffs between type I error and type II error (1 – power) for the design under consideration. Figure 10 shows the estimated power curves for the armor penetration example. We see that the power for velocity is much higher compared to the other model terms. The power for angle is particularly low, which is cause for concern because the subject matter expert suspects this is an important factor effect to estimate.

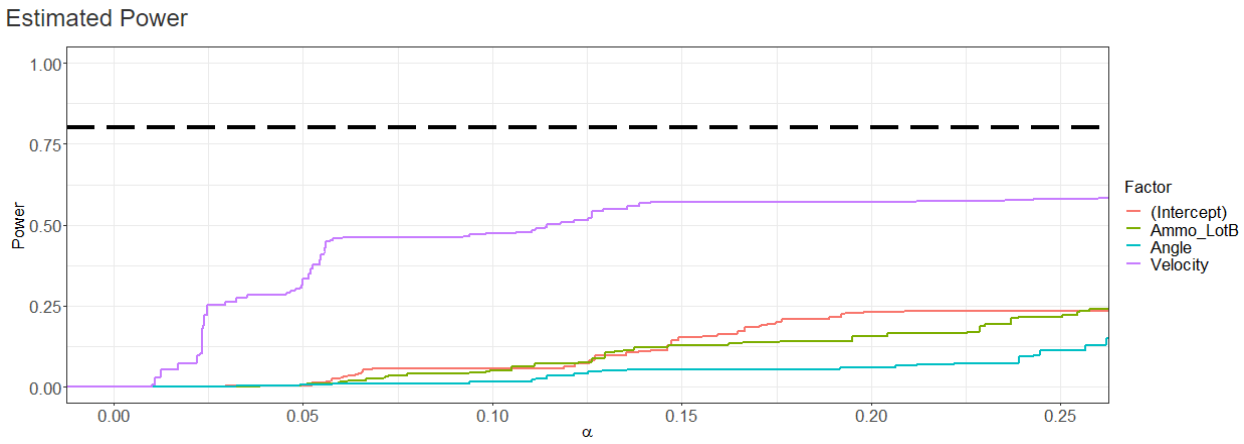


Figure 10: Estimated Power Across Values of Significance Level for Armor Penetration Example

P-Value Plot

This figure provides a visual display of the distributions of the p-values for each model term across all runs in the MC simulation. The figure shows a violin plot of the p-value for the hypothesis test associated for each model term in the logistic regression model. This figure provides insight into how frequently the model term is deemed statistically significant or not without specifying a specific significance level. Figure 11 shows the p-value distribution plot for the armor penetration example. Note that the p-values for velocity tend to be much lower compared to those for the other factors. This reflects the same information gleaned from the power plot.

Figure 11: p-value Distribution Plot for Armor Penetration Example

Standard Error Distribution Plot

This figure provides additional insight into the values of the standard error for the model coefficients to help assess the likelihood of separation when fitting the logistic regression model with data collected during the actual test execution. High values of standard error are indications that separation has occurred. The plot provides an option to graph the standard error on a log-scale to provide more informative graphs. Figure 12 shows the standard error plot for the armor penetration example. Note

that the standard error for the ammo lot tends to be low, while there were several iterations in the MC simulation for both angle and velocity where the standard error of the model estimates are quite large. This is an indication that separation is more likely for angle and velocity.

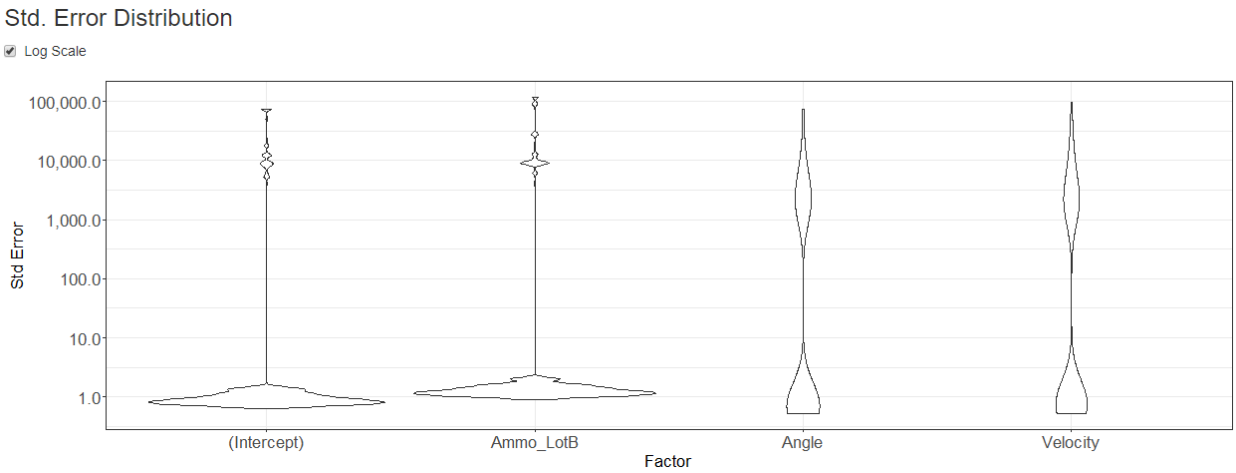


Figure 12: Standard Error Distribution Plot for Armor Penetration Example

Download Report

If you click the "Download Report" button, all of the described output are provided in a pdf document available for the user to use in any reporting documents.

Conclusion

This best practice introduced a new technique using MC simulation to estimate the power of a DOE when the response is binary. This method can provide valuable insights into the effectiveness of a test design for a binary response without relying on approximations to the normal distribution or design type. There is a disadvantage to this approach in that it requires eliciting critical information from the user: the estimated coefficient values for the logistic regression equation. This requirement puts a user in a "catch-22" situation: the user is executing a test to estimate the model coefficients, but to choose which design to run, they have to know *a priori* what those model coefficients are. This disadvantage can be mitigated by using the binary coefficient tool to help elicit information from a subject matter expert. Many practitioners will be able to provide ballpark estimates of a probability of success at particular test points compared to providing the value of the β_i values. In addition, we recommend doing sensitivity studies to estimate the power of the design. For example, obtain initial estimates for the model parameters and assess the properties of the design. Then repeat the design evaluation with different estimates of the model coefficients to assess how robust the design is to changes in the model parameters. This type of design evaluation is analogous to evaluating a design for a continuous response using differing SNRs and type I error rates.

We believe that while this method requires more information from the user, it can provide more informed assessments of the risk of a test design for a binary response. In the armor penetration

example presented throughout this best practice, we saw very low values of power for a 20-run test. Because this design is not an effective design, we considered alternative design options that increased the number of levels of the factors as well as the number of runs in the test to find a more appropriate test design. The app demonstrated throughout this best practice can be found at the following link: [INSERT LINK]

If you have any questions, please contact the STAT COE at COE@afit.edu.

References

Agresti, Alan. *Categorical Data Analysis*. 3rd ed., John Wiley & Sons, Inc., 2013.

Harman, Michael. "Test Design Comparison and Selection." Scientific Test and Analysis Techniques Center of Excellence (STAT COE), 29 Aug 2018.

Johnson, R. T. and Montgomery, D. C. "Choice of Second-Order response Surface Designs for Logistic and Poisson Regression Models." *International Journal of Experimental Design and process Optimization,* vol. 1, no. 1, 2009, pp. 2-23.

Montgomery, Douglas C. *Design and Analysis of Experiments.* 9th ed., John Wiley & Sons, Inc., 2017.

Natoli, Cory and Oimoen, Steve. "Classical Designs: Full Factorial Designs." Scientific Test and Analysis Techniques Center of Excellence (STAT COE), 1 Jun. 2019.

Natoli, Cory, Oimoen, Steve, and Burke, Sarah. "Categorical Data in a Designed Experiment Part 3: Logistic Regression." Scientific Test and Analysis Techniques Center of Excellence (STAT COE), 25 Aug. 2020.

Ortiz, Francisco (a). "Categorical Data in a Designed Experiment Part 1: Avoiding Categorical Data." Scientific Test and Analysis Techniques Center of Excellence (STAT COE), 23 Oct. 2018.

Ortiz, Francisco (b). "Categorical Data in a Designed Experiment Part 2: Sizing with a Binary Response." 18 oct 2018. Scientific Test and Analysis Techniques Center of Excellence (STAT COE), 18 Oct. 2018.

SAS STAT User's Guide. Version 15.1 2019
https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/statug/statug_logistic_details10.htm

Appendix: Binary Coefficients App

This appendix describes an additional tool developed by the STAT COE to help users estimate the model coefficients of the logistic regression model (β_i in Equation 1). This app is intended to be used in conjunction with the power app described in the main body of this document to find the coefficients of the logistic regression model. We demonstrate the features of this app with the armor penetration example previously described.

A critical input from the user is the estimated model coefficients to estimate the power of the test design. To determine these coefficients, we consulted with a subject matter expert who could give us ballpark estimates of the probabilities of penetration across the conditions of the velocity and angle. Figure 13 shows the factorial points for velocity and angle with the estimated probability of penetration at each corner point (left figure). The subject matter expert suspected that there could be up to a 5% difference between ammunition lots, and so we added 5% probability penetration at each factor condition to represent the effect of ammo lot B.

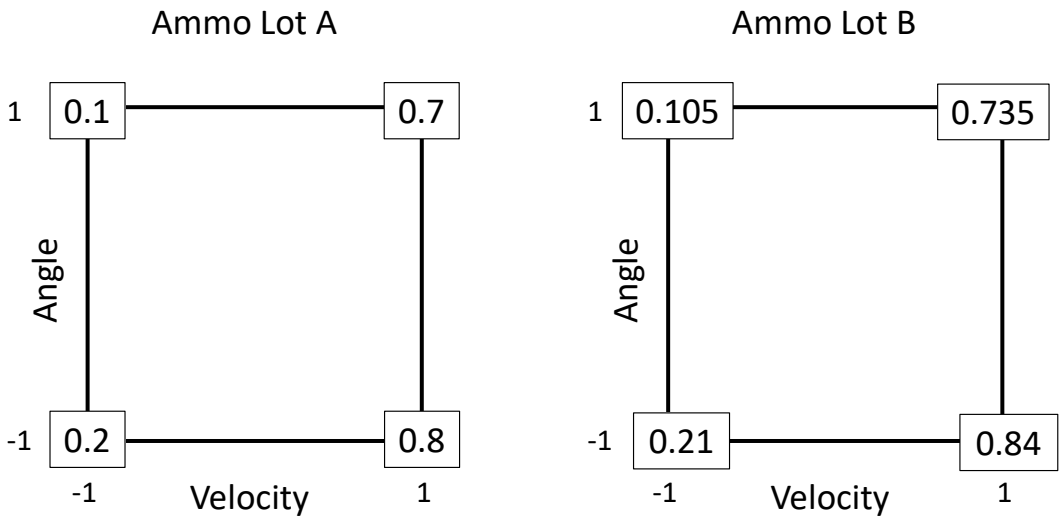


Figure 13: Armor Penetration Expert Probability Elicitation

We can now use this information from the subject matter experts in the Binary Coefficients App to estimate the model coefficients.

User Inputs

Use the side-panel to enter the details of the test design: the number of factors and the type of model. The app currently allows a maximum of five factors, assumed to be continuous. Specify if you wish to include two-factor interactions in the model by clicking the checkbox under "Select incorporated effects." Once all inputs are entered, click the "Update Table" button. Figure 14 shows the user inputs for the armor penetration example. Note that we do not include interactions in this example and we leave the factor units coded between -1 and 1.

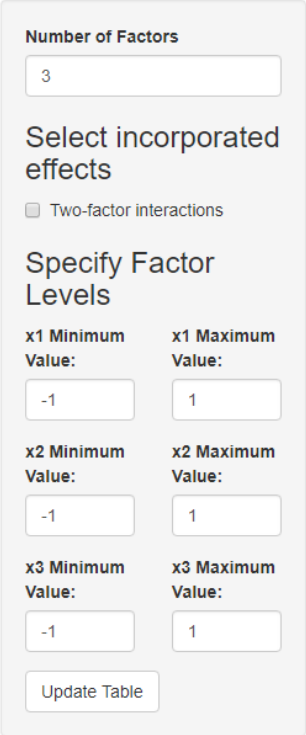


Figure 14: Binary Coefficients App User Inputs

After clicking the "Update Table" button, the table in the main panel of the app shows all possible test conditions where all factors are either at the lowest or highest value. The final column in this table shows the probability of success for each factor level combination. We enter in the information elicited from the subject matter expert in this column. Ensure that the definition of a success is clearly understood for the subject matter expert. The default values for the probability of success are set to 0.5, as seen in Figure 15a. Users should update the probability of success for every condition to the best of their knowledge.

x1	x2	x3	Probabilities
-1.00	-1.00	-1.00	0.50
1.00	-1.00	-1.00	0.50
-1.00	1.00	-1.00	0.50
1.00	1.00	-1.00	0.50
-1.00	-1.00	1.00	0.50
1.00	-1.00	1.00	0.50
-1.00	1.00	1.00	0.50
1.00	1.00	1.00	0.50

a)

x1	x2	x3	Probabilities
-1.00	-1.00	-1.00	0.20
1.00	-1.00	-1.00	0.80
-1.00	1.00	-1.00	0.10
1.00	1.00	-1.00	0.70
-1.00	-1.00	1.00	0.21
1.00	-1.00	1.00	0.84
-1.00	1.00	1.00	0.11
1.00	1.00	1.00	0.74

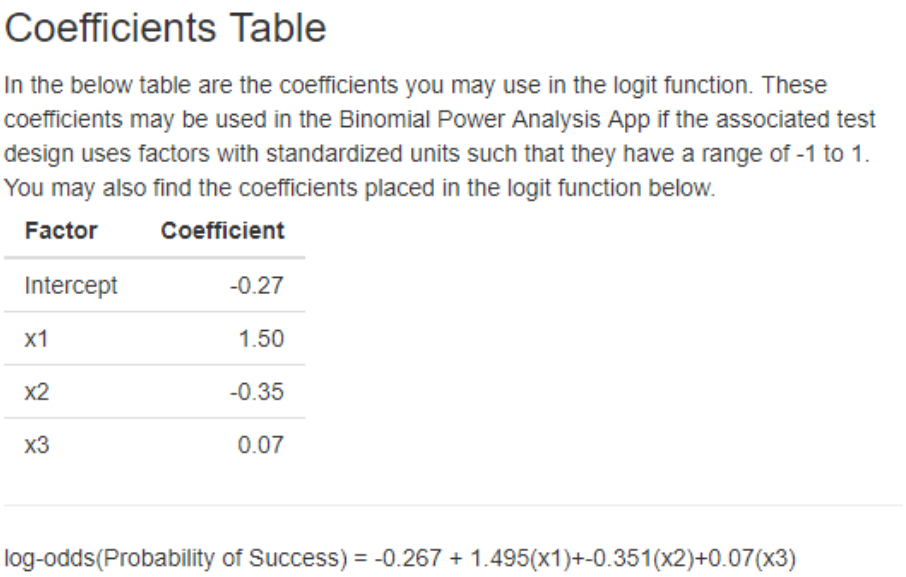
b)

Figure 15: Binary Coefficients App a) Default entries; b) entries for the armor penetration example

Figure 15b shows the probabilities elicited from the subject matter expert for the armor penetration example. Note that the app restricts probabilities to be between 0.01 and 0.99 to obtain stable estimates of the model coefficients. Click the "Update Coefficients" button.

Binary Coefficients App Output

The "Coefficients Table" shows the values of the model coefficients to use in the logistic regression model. The app uses a solver to determine these values, and assumes the model for the binary responses uses the logit function (as seen in Equation 1). The values of the coefficients in this table can be used in the binary power app, as was seen in Figure 5. The app also shows the log-odds of the probability of success listed in an equation form. This is simply the linear component in the logistic regression equation. Figure 16 shows the coefficients table for the armor penetration example; these values were used in the example shown in the main body of this best practice to estimate the power of the test design.



Coefficients Table

In the below table are the coefficients you may use in the logit function. These coefficients may be used in the Binomial Power Analysis App if the associated test design uses factors with standardized units such that they have a range of -1 to 1. You may also find the coefficients placed in the logit function below.

Factor	Coefficient
Intercept	-0.27
x1	1.50
x2	-0.35
x3	0.07

log-odds(Probability of Success) = -0.267 + 1.495(x1)+-0.351(x2)+0.07(x3)

Figure 16: Coefficients Table for Armor Penetration Example

Current limitations of this app are listed below. Current work is underway to address some of these limitations.

- Assumes factor levels are coded (-1 to 1). If your test design is in actual units, evaluate the test design using coded units to seamlessly use both apps.
- Assumes model for 1 to 5 factors
- Assumes all factors are continuous
- Allows main effect and two-factor interaction models only
- Restricts probability entries to be between 0.01 and 0.99 to obtain more stable coefficient estimates.
- Assumes the logit function in the model for the binary response