4-17-2017

Logistic Regression Slope Study

Michael J. Kist
mjk6375@rit.edu

R·I·T

Logistic Regression Slope Study

by

Michael J. Kist

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Industrial and Systems Engineering.

•

Department of Industrial and Systems Engineering

Kate Gleason College of Engineering

•

Rochester Institute of Technology

Rochester, NY

April 17, 2017

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

KATE GLEASON COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NY

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree thesis of Michael Kist
Has been examined and approved by the
Thesis committee as satisfactory for the
Thesis requirements for the
Master of Science degree
Approved by:

Dr. Rachel Silvestrini **Date**
Thesis Advisor

Dr. Scott Grasman **Date**
Committee Member

Abstract

Logistic regression is a valuable statistical tool used to model the probability of a binary response variable as a function of one or more input variables. The goal of this thesis research is to develop a better understanding of how the coefficients of a logistic regression model influence the probability of a response. Typically, the odds ratio is used for this, but this research focuses on the steepness of logistic curve near the median quantile. In order to study this, a web application using R Shiny was developed to simulate a logistic regression function based on a single continuous input variable. The web application allows a variety of inputs to be manipulated, including sample size, noise structure, amount of noise, and actual parameter values. An example using the NASA O-Ring data is illustrated as motivation and discussion.

Acknowledgements

I would like to thank my advisor Dr. Rachel Silvestrini for her tremendous support and guidance during my time at RIT. Her support has allowed me to grow as a scholar and as a person and I am truly grateful for all she has done for me.

I would also like to thank the department of Industrial and Systems Engineering, headed by my committee member Dr. Scott Grasman for all the opportunities they have presented to me. It has been an absolute pleasure perusing my degree in this department.

Finally, I would like to thank my family, specifically my parents, for their endless support and encouragement. Without them, these opportunities would not have been available to me and for that I am forever grateful.

Table of Contents

1. Introduction

Logistic regression is a mathematical technique that is used to quantify the relationship between a binary response variable and one or more input variables. Applications of logistic regression in the literature highlight valuable insight based on the logistic regression models developed. For example, Tan et al. (1993) utilize logistic regression to analyze the probability of successful coronary angioplasty as a function of several input variables. Lu et al. (2000) sought use logistic regression to model the value, C_{50}, which is the medication dosage that correlates to a 50% probability of successful effect. Their paper indicates accurate data when compared to the population responses.

Another example of logistic regression application is for the O-Ring success model as a function of ambient temperature. After the tragic failure of the 1986 Challenger space shuttle, the National Air and Space Administration (NASA) disclosed that the cause of the explosion was O-Ring failure caused by abnormally low temperatures prior to launch. The reason the launch continued was due to lack of agreement on the interpretation of the given data as the previously recorded 23 launches did not occur at such low temperatures. Maranzano and Krzysztofowicz (2008) outline this problem and discuss the challenge the NASA engineers faced due to the extrapolation necessary to draw conclusions using the logistic regression model as well as other comparable models.

Not considering extrapolation issues, interpretation of the relationship between the model inputs and the binary response using logistic regression can be difficult. For comparison, consider the linear regression model. Linear regression is a statistical technique used to model the relationship between one or more input variables and a continuous response variable. One of the most useful characteristics of linear regression, is that the relationship between the input and response variable can be easily interpreted based on the value of the estimated slope parameter. As the model complexity increases and/or a different mathematical model is used, the interpretation of the relationship becomes harder.

To illustrate the interpretation of a linear regression model, consider modeling the elasticity of an O-Ring, which is a continuous random variable response, as opposed to the success or failure of an O-Ring (the binary response). Elasticity is an essential characteristic of an O-Ring is how much the ring will deform under certain environmental conditions, such as temperature. Assume we fit a linear model to observe how the elasticity, *E,* changes as a function of the ambient temperature, *F*. An example of a linear regression model is:

$$y_E = \beta_0 + \beta_F x_F + \varepsilon \qquad (1)$$

In this case, y_E represents the response variable or the value we are trying to predict. The coefficient β_0 represents the intercept coefficient, which is the value of y_E we would expect when x_F is 0. β_F is the slope coefficient associated with x_F. The variable x_F is the input variable or the independent variable for which we are using to predict y_E. Finally, ε represents any anticipated random error or noise present in the model.

After data collection, we fit the model, and the equation becomes:

$$\hat{y}_E = \hat{\beta}_0 + \hat{\beta}_F x_F \qquad (2)$$

In this form, the estimated amount of elasticity change we can anticipate per 1 unit of temperature change, is denoted by $\hat{\beta}_F$. If the value of $\hat{\beta}_F$ is 0.05, then we know for every degree of temperature increase, we can expect a 0.05 unit increase in elasticity. When using logistic regression, the response is an event probability and not a specific characteristic and the model is no longer linear, thus the interpretation become more complicated.

The logistic regression equation, using the logit link function, is as follows:

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \qquad (3)$$

Given the structure of the logistic regression equation in Equation 3, it is difficult to interpret exactly what kind of influence the coefficient β_1 will have on our response. One common approach is to use the odds ratio (OR). Hosmer et al. (2013) show the algebra associated with the OR and how we can use it to help our interpretation. The OR is determined by the following equation:

$$OR = e^{\widehat{\beta}_1} \qquad (4)$$

Here, we see the OR simplified as a function of our β_1 coefficient. We interpret this as follows; if our OR is equal to 1, x_1 has no influence on our response probability. If OR is greater than 1, increasing the variable x_1 will increase the probability of the binary event occurring. And if OR is less than 1, increasing the variable x_1 will decrease the probability of the binary event occurring. The challenge lies in the fact that it is very difficult to understand and visualize the slope or rate of change that is associated with a given odds ratio, and relating the odds ratio to the span of the input variable is not straightforward. Understanding how a change in the β

coefficients in logistic regression influence the response is not as easily observed as it is in linear regression. The primary goal of this thesis is to research methods that allow a better understanding of how changes in specific input variables influence the rate of change of the binary response.

While the primary objective of this thesis research is to study the relationship between logistic regression model parameters and predicted probabilities of response, a secondary goal was to develop a software that can be used by practitioners reproduce these results and conduct their own research regarding the logistic regression function. Section 2 provides a brief literature review of relevant sources to the applied research. Section 3 outlines the methodology and test procedure followed to conduct the research. Section 4 includes a summary of the results of the analysis. Section 5 presents conclusions and future work related to this research.

2. Literature Review

In this section, peer reviewed articles relevant to the research conducted in this thesis are discussed. Several statistical tools and techniques are applied to the research conducted in this thesis. Journals that have applied these techniques were studied to aid in our application. The discussion will also include the use of confidence intervals, threshold values, and interpretation of logistic regression results.

When studying the behavior of confidence intervals with respect to linear modeling such as the linear logistic model, it is important to understand how the results may differ between data sets. Van Ewijk and Hoekstra (1994) saw many different shapes of response curves when conducting their curvature analysis. Comparisons were drawn on two commonly used methods of generating confidence intervals for the linear logistic model. The curvatures of 55 unique data sets were observed based on fits generated through Gauss-Newton method. The results found that the confidence intervals of the linear logistic model proved difficult to observe given its high intrinsic curvature. This was an interesting conclusion for this research as the linear logistic model is a closely related to the logistic regression model that we will discuss.

Hurwitz and Remund (2014) also test the application of confidence interval calculations to multiple levels of threshold and compare them across separate logistic regression models. Their study applies to the extreme, but reasonable thresholds, including 10%, 20%, 50%, 80%, and 90% probability of success. Their results indicate consistent confidence interval results across all thresholds, measuring less than 1% deviation between the actual simulated results and the applied confidence levels.

Kist and Silvestrini (2016) analyzed use of confidence intervals in conjunction with the decision threshold values to their combined influence on the predictive capability of logistic regression. The research allowed for the development of a theoretical ‘grey area’ in which observations of high uncertainty fell. The grey area is a region of high uncertainty that is placed around the specified threshold value in which a response of 1 or 0 is highly likely to occur. Several data and model characteristics were tested to understand their influence on the grey area and the percentages of error in classification rates. This thesis extends this analysis further by conducting research on how quickly the probability of success changes within the grey area.

The following figures and descriptions will illustrate the process of grey area development, which was originally derived in the previous work of Kist and Silvestrini (2016). A graphical depiction of binary response data and a single input variable can be seen in Figure 1, where the input values line the X axis and the Probability of Success is shown on the Y axis.

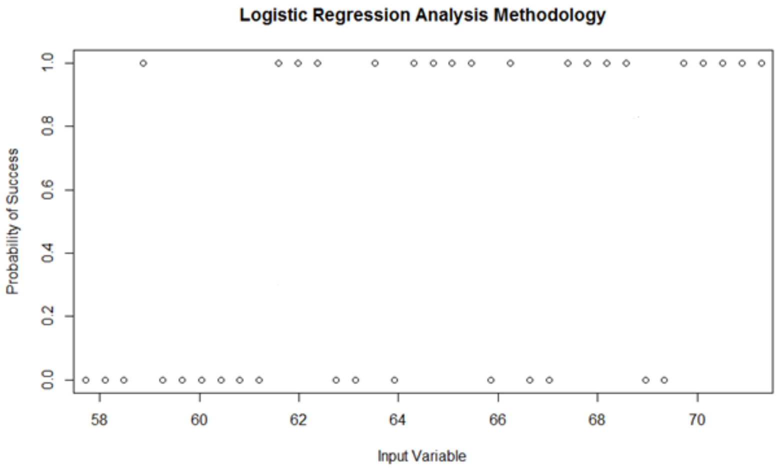


Figure 1: The raw data developed is plotted as either 0’s representing failure, or 1’s representing success, based on the input variable in question.

Once the data has been plotted, we use the generalized linear model or GLM function to plot a link logit binomial fit to the data set. This is what tells us what our simulated data’s logistic regression coefficients are. Using this GLM fit, we can then plot the S-Curve over the binary data, to show how the fit is represented with respect to the input variable. Figure 2 shows the binary data with the logistic regression S-Curve plotted as well. Equation 5 represents the model fit associated with the logistic regression curve shown in Figure 2.

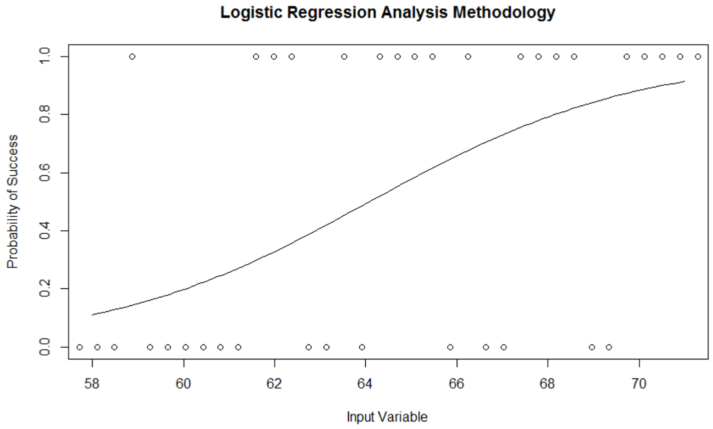


Figure 2: The same data with the corresponding logistic regression curve that represents the relationship of the input variable and the response probability.

$$P(x) = \frac{1}{1 + e^{-(-15.04 + .23x)}} \qquad (5)$$

After plotting the logistic regression curve, we generate upper and lower confidence bands based on the stand error of the model fit. The confidence intervals are plotted as well to aid in the visualization. This can be seen in Figure 3, where 95% confidence bands have been applied.

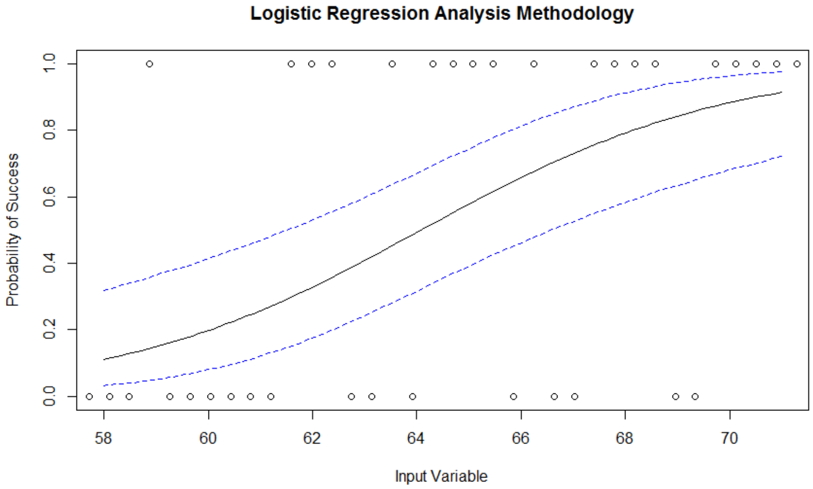


Figure 3: Confidence intervals applied to the logistic regression curve to show the potential upper and lower 95% confidence bands of the true fit.

The third step in the analysis is to determine the input value that corresponds to the specified threshold. In this example, we observe the 50% threshold and therefore we use the regression fit to find the input variable corresponding to a 50% probability of success. Once this value is calculated, we plot a vertical line connecting it to the upper and lower confidence intervals. This is shown in Figure 4. This gives us points A and B which are our upper and lower limits of our true probability threshold given 95% confidence bands.

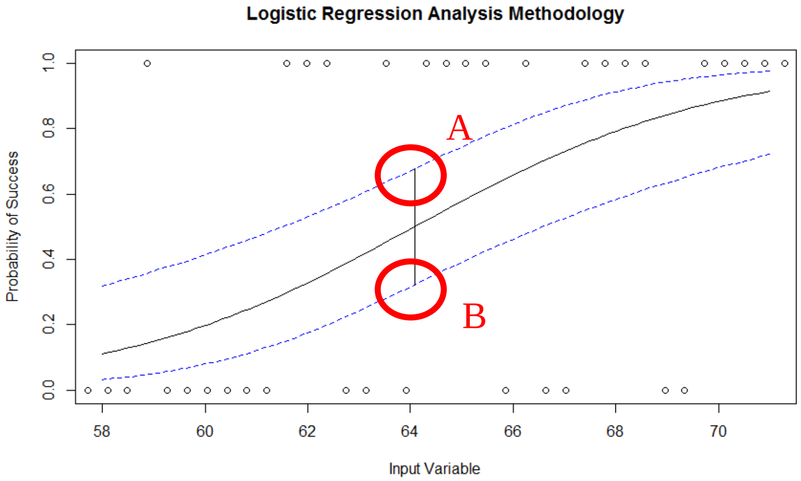


Figure 4: The 50% threshold line is drawn to show the upper and lower probability of success at the input variable associated with the 50% probability of success.

Step four in our analysis takes points A and B and finds their corresponding probability of success values. In this example, the probability of success for point A is approximately 0.70 and for point B it is approximately 0.30. Precise values are calculated for the analysis. We take these two values and finds the corresponding input value of the initial logistic regression model. To show how this translates graphically, we plot a connecting line from points A and B horizontally to the initial logistic regression curve. This is shown in Figure 5. This gives us point C and D, which will be used to calculate our logistic regression model's slope value.

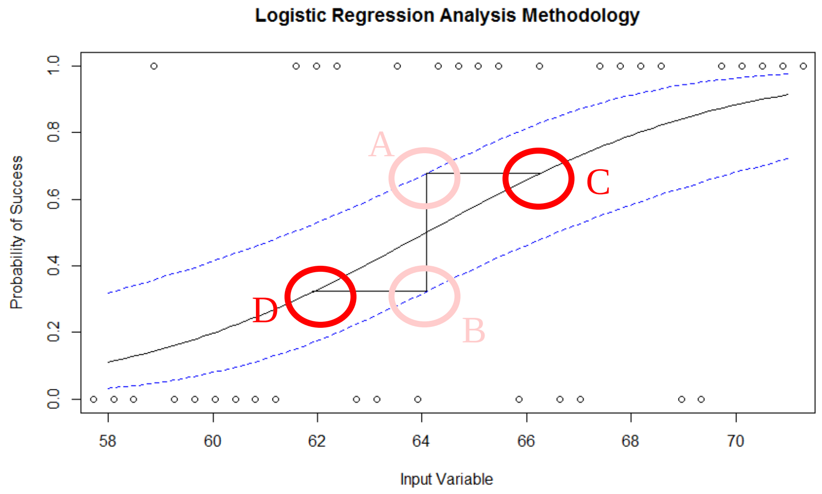


Figure 5: The upper and lower probability values are connected horizontally to the logistic curve to show the points used to calculate the logistic regression middle quantile slope.

While Points C and D provide us with some of the more important information of the analysis, there are some additional steps taken to collect information on the output. The next step outlines the entire grey area which encompasses the area between points D and C as well as the 95% confidence bands. This is shown in Figure 6. The size of the grey area is another response that is important to our analysis.

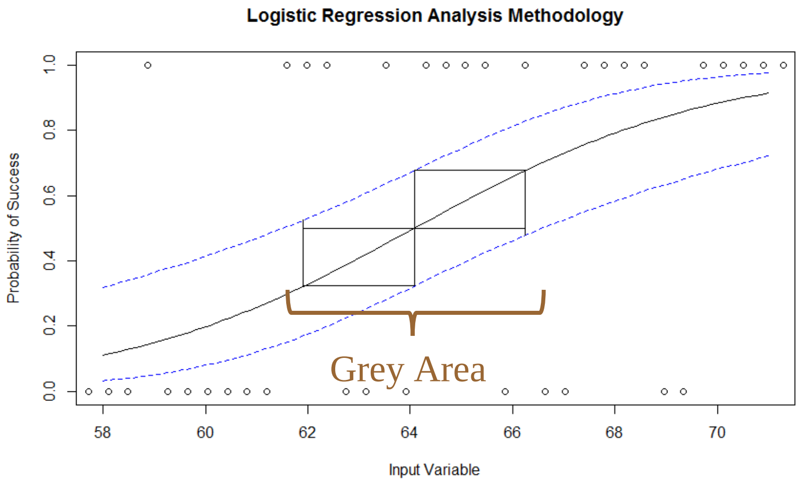


Figure 6: The connected upper and lower limits are used to develop the grey area or area of uncertainty at the middle quantile.

A concern with statistical analysis in general is the issue of misinterpretation of the results, which is a key factor we are considering when looking at the basic logistic regression model. Bender et al. (1996) discusses the issues of improper procedure and interpretation in the growing number of medical papers that utilize logistic regression. Bender et al. (1996) warn of the lack of the assessment of goodness of fit in recent studies. Stoltzfus (2011) discusses that while goodness of fit is a critical part of the interpretation, it is also essential to run diagnostic statistics to ensure model integrity before drawing conclusions on the quality of a model. These aspects are essential details to consider when discussing the logistic regression model.

When performing analysis on the logistic regression model, proper experimentation with varied coefficients should be considered critical to the results. Kerns (2016) considers a series of five coefficient sets for conducting simulation analysis on confidence bands applied to them. This set of coefficients covers several different potential scenarios faced in logistic regression based on how aggressively the probability increases or decreases. This gives an all-encompassing perspective on how certain responses may vary under different circumstances. We also strive to produce results based on multiple coefficient pairings in simple logistic regression and the model coefficients suggested by Kerns (2016) will be used in our analysis.

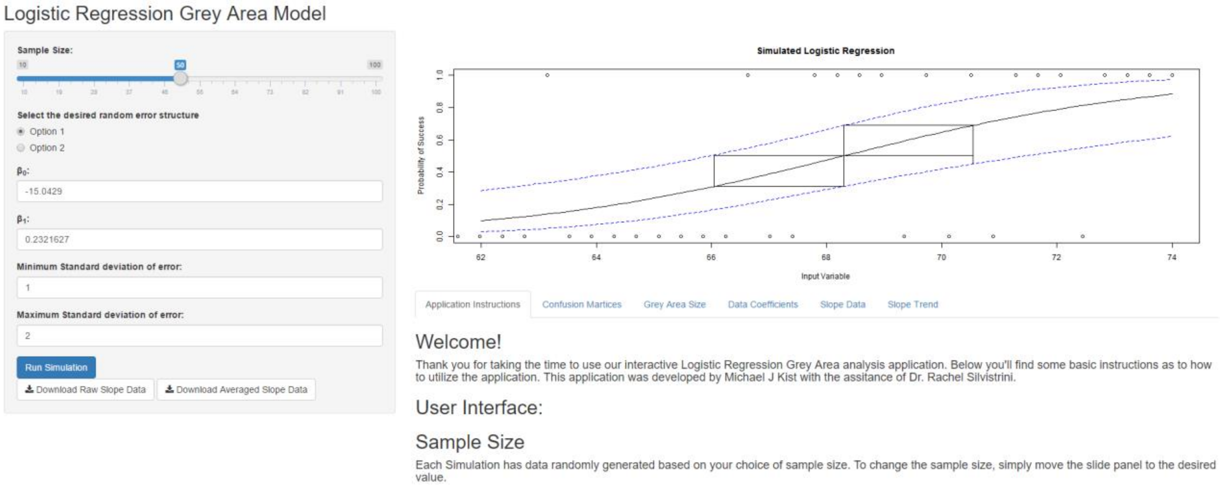
3. Methodology

In the following section, we will outline the steps taken and methods followed to replicate this analysis. We will discuss the specifics of our simulations and data generation and address the process of collecting and processing that data using an application developed for this research. The programming language R will be used to generate all experimental data and collect results. Specifically, the RShiny package was used to develop an interactive web application for replicating results and practitioner use. The details of this application will be discussed.

We use this web application for this thesis to gain a better understanding of the relationship between our input variables and the slope of the grey area, detailed in section 2. The slope of the grey area is represented by the upper and lower extremes of our area of uncertainty. We worked to develop a mathematical representation of this slope relative to the variables we studied.

3.1 Shiny Application

Two web application were developed. The first was developed so that a practitioner can fit and study a logistic regression model based on existing data. The second was developed to simulate data from a logistic regression model. In the second web application, the user can generate and view high level results of the simulation or extract their complete set of data for more specific analysis. Figure 7 displays the current layout of the second shiny web application, which was the primary tool used for this research and will be discussed in this section.



Sample Size: 50

Select the desired random error structure

● Option 1
○ Option 2

β_0: -15.0429

β_1: 0.2321627

Minimum Standard deviation of error: 1

Maximum Standard deviation of error: 2

Welcome!

Thank you for taking the time to use our interactive Logistic Regression Grey Area analysis application. Below you'll find some basic instructions as to how to utilize the application. This application was developed by Michael J Kist with the assitance of Dr. Rachel Silvistrini.

User Interface:

Sample Size

Each Simulation has data randomly generated based on your choice of sample size. To change the sample size, simply move the slide panel to the desired value.

Figure 7: RShiny Web Application Main view that is shown upon opening the application.

When the application is opened, it accepts several input values. The inputs include: threshold value, sample size, confidence interval size, error structure, coefficient values, upper and lower range values, and the standard deviation of the error. Once the user enters in their desired values and presses the "Run Simulation" button, they will be provided with a graph and several descriptive statistics of their simulated data set. They also have the option of downloading the simulated data set.

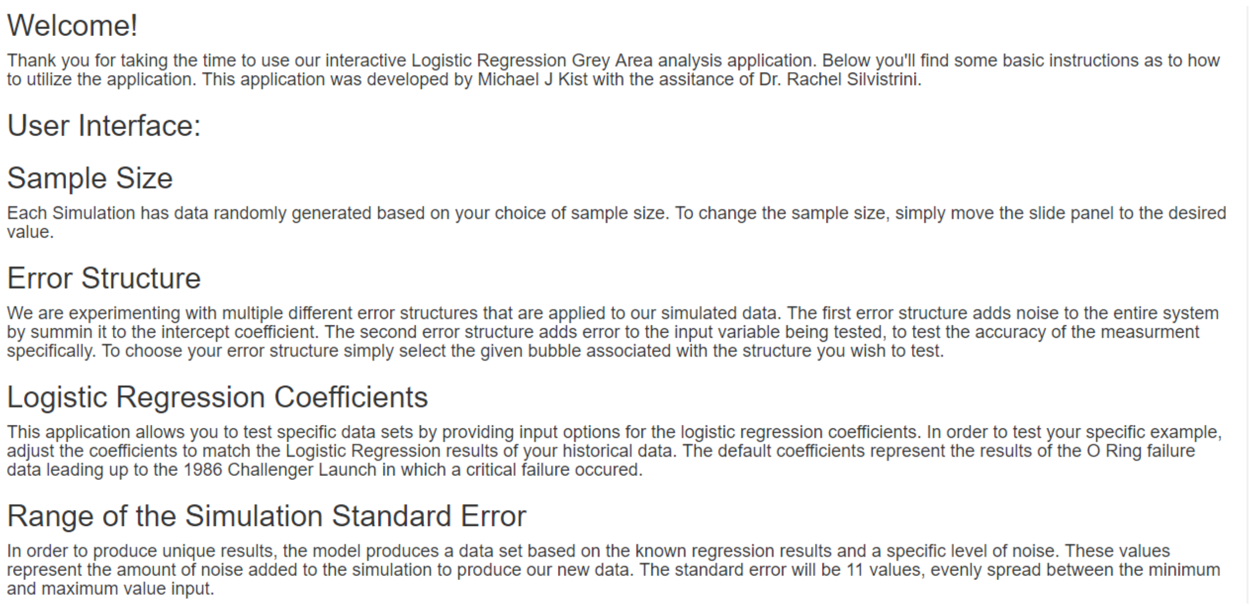
After the data is generated, a logistic fit using the logit link is performed. Once the model is fit, we operate several functions to plot the fit so that the user may visualize how their data interacts with the input variable they wish to analyze. Along with the logistic function plot, we also plot upper and lower confidence intervals that are relative to the fit and based on the desired input percentage.

When generating the grey area, the intersections of the confidence intervals and the threshold values needs to be calculated and stored so that we may display a visualization of the grey area. Segments are added to the plot and connect the upper and lower intervals to the calculated grey area limits. These limits are used to calculate the numeric statistics that are presented to the user.

The user has three tabs they may navigate while analyzing their data. These different tabs are explained in the opening view with a list of instructions on how to utilize and interpret the application and its output.

Overall this web application will be a useful tool to continue to develop as the research progresses. It will allow people who take interest in our research to test it themselves and run experiments on their simulated data. As the research advances the application will be updated to include any additional analysis being done to the model and its relationships of interest.

Figure 8 shows the welcome page, which provides details for the user on the contents of the web application and how it is used.



Welcome!

Thank you for taking the time to use our interactive Logistic Regression Grey Area analysis application. Below you'll find some basic instructions as to how to utilize the application. This application was developed by Michael J Kist with the assitance of Dr. Rachel Silvistrini.

User Interface:

Sample Size

Each Simulation has data randomly generated based on your choice of sample size. To change the sample size, simply move the slide panel to the desired value.

Error Structure

We are experimenting with multiple different error structures that are applied to our simulated data. The first error structure adds noise to the entire system by summin it to the intercept coefficient. The second error structure adds error to the input variable being tested, to test the accuracy of the measurment specifically. To choose your error structure simply select the given bubble associated with the structure you wish to test.

Logistic Regression Coefficients

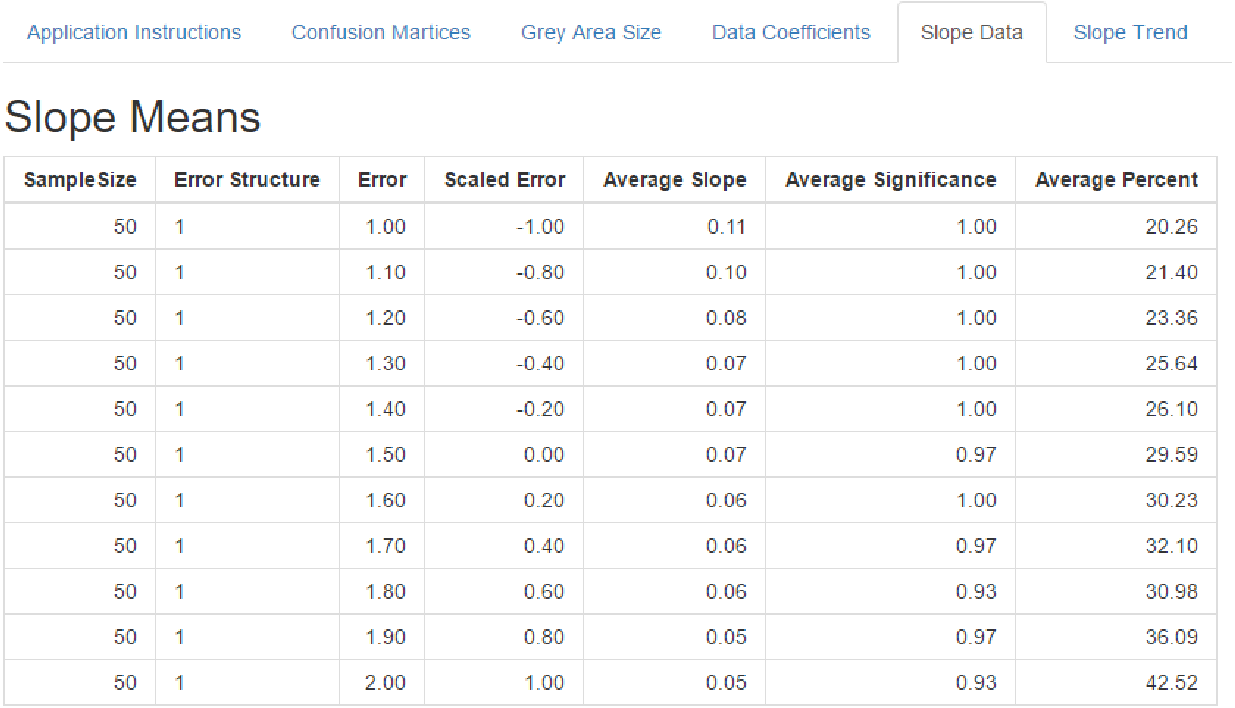
This application allows you to test specific data sets by providing input options for the logistic regression coefficients. In order to test your specific example, adjust the coefficients to match the Logistic Regression results of your historical data. The default coefficients represent the results of the O Ring failure data leading up to the 1986 Challenger Launch in which a critical failure occured.

Range of the Simulation Standard Error

In order to produce unique results, the model produces a data set based on the known regression results and a specific level of noise. These values represent the amount of noise added to the simulation to produce our new data. The standard error will be 11 values, evenly spread between the minimum and maximum value input.

Figure 8: Instructions shown in the application to aid the user with their experience using the application.

Figure 9 provides a snap shot of one of the tabs available in the web application. The application has been adapted throughout the research to adequately fit the model and the code that it represents can be found in Appendix 1. The table shown in Figure 9 provides a results summary of the simulations that were run. The first four columns represent the model parameters that were set. The columns to the right outline results such as the slope of the grey area, the percent of the input range that the grey area consumes.



Slope Means

SampleSize	Error Structure	Error	Scaled Error	Average Slope	Average Significance	Average Percent
50	1	1.00	-1.00	0.11	1.00	20.26
50	1	1.10	-0.80	0.10	1.00	21.40
50	1	1.20	-0.60	0.08	1.00	23.36
50	1	1.30	-0.40	0.07	1.00	25.64
50	1	1.40	-0.20	0.07	1.00	26.10
50	1	1.50	0.00	0.07	0.97	29.59
50	1	1.60	0.20	0.06	1.00	30.23
50	1	1.70	0.40	0.06	0.97	32.10
50	1	1.80	0.60	0.06	0.93	30.98
50	1	1.90	0.80	0.05	0.97	36.09
50	1	2.00	1.00	0.05	0.93	42.52

Figure 9: Summarized results and figures are shown in the tabs to provide simple results and representations of the simulated data.

3.2 Simulation

The flow chart in Figure 10 outlines the generic process of the simulation, data collection, and analysis that will be utilized in this research.

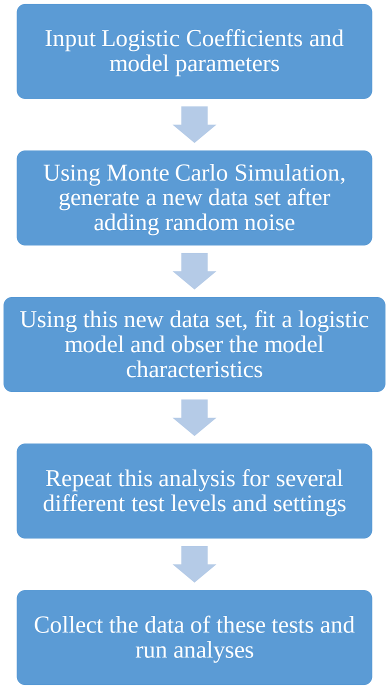

Input Logistic Coefficients and model parameters
Using Monte Carlo Simulation, generate a new data set after adding random noise
Using this new data set, fit a logistic model and obser the model characteristics
Repeat this analysis for several different test levels and settings
Collect the data of these tests and run analyses

Figure 10: Simulation and Data Collection Flow Chart

Unlike linear regression, in which an error term is represented as an added value to the regression equation, error terms in a logistic regression are not straightforward. The model used for the original grey area research utilized the error structure that is shown in Equation 6. The error term is denoted by ε and is assumed to be NID $(0, \sigma^2)$. In addition to the first equation, Equation 7 represents an alternative method of error incorporation that will also be studied.

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x + \varepsilon)}} \tag{6}$$

$$P(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1(\varepsilon + x))}} \tag{7}$$

Throughout this thesis, we will reference the methods of error incorporation shown by Equation 6 and 7 as Error Structure 1 and 2, respectively.

The key relationship that will be analyzed regarding the varying methods of error application and the slope variance is the influence on the steepness of the logistic regression curve at the middle quantile. Measuring and understanding how these variables interact will be critical to understanding the capabilities of the model and its overall usefulness. Observing the size of the grey area under a variety of conditions will allow us to better understand the situations in which this model will be applicable.

The model developed will simulate data based on a current set of data such as the O-Ring data with one of the error formulas. When the data is generated and the confidence interval is applied to the chosen threshold, the upper and lower bounds of the grey area will be recorded. This will give us ranges that can be used to analyze the impact on the range of the input variable. Further detail on the data collection will be outlined in the following section.

3.3 Data collection process

In this analysis of the logistic regression model, we consider several different variables and how they influence the results of the simulation. Those variables include: Sample size, logistic regression coefficients, method of error incorporation, and standard error of the simulated data.

To be as comprehensive as possible, tests were conducted across both error structures to observe their influence on the measured responses. The following tables outline the levels of the variables tested for each data set. Table 1 indicates the four sets of logistic coefficients, including a brief description of the type of curve produced. The first three coefficient sets were presented in Kerns (2016).

Table 1: This outlines the series of logistic regression coefficients used in our simulation to model unique logistic regression curves.

Coefficients	Description
[-2, .3]	Represents an S-curve with a slowly increasing slope.
[0, 1.5]	Represents an S-Curve with a moderately increasing slope.
[2, 5]	Represents and S-Curve with a fast increasing slope.
[-15, .23]	Represents the 1986 Challenger O-ring temperature data.

Table 2 shows the breakdown of the test levels that will be applied for each set of coefficients. These were the final experimental levels used to analyze the variables that were found to have significant interaction with the response after the preliminary analysis. A threshold level of 0.5 and a confidence level of 95% were used for this analysis. It should be noted that additional levels of standard error will be tested at larger sample sizes. As the sample size grows, the minimum standard error can decrease without providing pure separation between the binary response outputs. In Error Structure 1, the minimum error analyzed is .25 and in Error Structure 2 the minimum is .5.

Table 2: This outlines the different test levels and model characteristics used for each Error Structure during the simulation testing.

Test Levels										
Error Structure	1					2				
Sample Size	10	15	20	50	100	10	15	20	50	100
Standard Error [min, max]	[1, 5]					[3, 15]				

Another portion of our analysis is to determine if there is a present significance between the two extremes of the data set. While the user can determine the extreme points which we compare, the largest and smallest input value of the simulated data are the default values. We calculate the probability and the lower probability bound of the higher extreme point and the upper probability bound of the lower extreme. If the bounds overlap, we can conclude with 95% confidence that there is no significant difference between the probabilities of success at the extreme points.

4. Results

In this section, we provide a summary of the results we have generated. First, we will reveal some approximate slope results that were collected through our Monte Carlo simulation. These slope values are what we seek to better understand with respect to their relationship with the model factors. Secondly, we will begin to explain some of the key factors used in our model and show their relationship with the slope values. This includes factors such as sample size, coefficients, and standard error of the simulation. Next, we discuss whether statistical significance was observed between the confidence bands of the 10% and 90% threshold values. We wish to discover which factors often lead to significantly different results and the antithesis. Finally, we conclude the section with a linear regression analysis of the slope values for some of the experiment combinations. Additionally, throughout this section we will discuss the differences between Error Structures 1 and 2.

The primary response in this experiment was the slope of the regression curve at its steepest point. Since this slope is relative to the grey area, it gives us a relative perspective on the size of the uncertainty region. With proper scaling, the value of this slope can be calculated and allow us to make effective conclusions about specific data sets. While there is no optimal slope value, it is an important metric to help us further understand what we can expect from certain data sets.

4.1 $\boldsymbol{\beta}$ Coefficient Values

The work of Kerns (2016) outlined a series of logistic regression coefficients that generate very specific outputs. These coefficient values are useful because they target specifically the rate at which the curve grows.

In Table 3, we have outlined the average slope of the S-Curve for each coefficient value at varying sample sizes and levels of standard error for Error Structure 1. In Error structure 1, we see each set of coefficients acting very similarly. At low error values the peak slope values are observed and then they reduce quickly as error increases. We observe that the coefficients have a clear relationship with the value of the slope. As anticipated, the fast-increasing coefficients showed the steepest slope and the slow increasing coefficients provide the smallest slopes.

Table 3: The average slope values for each set of characteristics applied to Error Structure 1.

Beta	Sample Size	Error Structure 1		
		σ_{Low}	σ_{Med}	σ_{High}
[-2, .3]	20	1.83	0.58	0.22
	50	4.47	0.47	0.25
	100	6.00	0.83	0.36
[0, 1.5]	20	73.81	17.24	11.52
	50	142.31	19.9	13.59
	100	168.41	20.3	13.26
[2, 5]	20	502.94	122.48	66.04
	50	1241.12	202.49	80.07
	100	1650.45	243.47	79.01

Slow increasing S-Curves with low error are comparable in slope to medium increasing S-Curves with high error. The results also suggest, unsurprisingly, that as our error values increase to higher values, the slope will asymptotically approach 0. This means that when there is a very small odds ratio or a weak relationship between x and y, as error increases, we fail to see any relationship at all.

Next, we examine the results for Error Structure 2, shown in Table 4. What the data suggests is that the slope results have much more variability in this error structure and do not change as dramatically based on the slope coefficients. There is still a noticeable slope increase when we increase the steepness of the S-curve and the sample size of the simulation. We also observe a similar trend where, as error increases, slope decreases in almost all cases.

Table 4: Table summarizing average slope values for the different test levels.

S-Curve Increase	Sample Size	Error Structure 2		
		σ_{Low}	σ_{Med}	σ_{High}
[-2, .3]	20	2.00	0.70	0.40
	50	5.60	0.88	0.32
	100	7.57	0.95	0.38
[0, 1.5]	20	11.71	5.78	9.07
	50	9.83	5.65	2.95
	100	12.40	5.13	2.85
[2, 5]	20	32.43	28.51	39.62
	50	13.03	17.27	15.94
	100	23.04	18.22	12.26

The data produced very different results from Error Structure 1, which we attribute to the higher error values. Reducing the error lead to limitations with our experimentation. In our simulation, at very low sample sizes, 20 and lower, separation occurs when the error was too low. Separation means that at a specific input value, every input above that value results in the same outcome and every data point below shows only the opposite outcome. Thus, when separation occurs, the coefficients cannot be fit using maximum likelihood estimation.

To account for this, the following adjusted analysis will only be conducted on experiments with sample sizes of 50 and 100. With these sample sizes, we could lower our standard error in our simulation and produce more accurate results for both error structures. These results are shown in Table 5. The error minimum and maximum for Error Structure 1 was changed from [1, 5] to [.25,5] and for Error structure 2 they shifted from [3,15] to [.5,5].

Table 5: Slope Results for larger Sample Sizes with less error.

S-Curve Increase	Sample Size	Error Structure 1			Error Structure 2		
		σ_{Low}	σ_{Med}	σ_{High}	σ_{Low}	σ_{Med}	σ_{High}
[-2, .3]	50	20.52	0.76	0.18	24.41	6.76	2.38
	100	52.86	1.14	0.36	78.71	8.64	3.22
[0, 1.5]	50	926.86	31.31	12.45	209.67	11.79	6.55
	100	1874.06	32.19	11.42	305.85	12.66	4.34
[2, 5]	50	4741.8	253.62	56.32	306.44	20.79	23.37
	100	12445.7	258.73	91.35	267.75	25.59	21.83

The adjusted experiment revealed that the initial error values were not entirely accurate, as they were skewed to account too heavily for low sample size simulations. We now see a much clearer and more consistent trend through both Error Structures based on the coefficients of the simulated data. The error values used were as low as they could feasibly be without throwing separation errors.

What is interesting about these slope results, is that for slow ascending S-Curves, when error is added to the input variable, it produces higher slope values than when error is added to the whole system. The opposite is true for medium and fast ascending coefficients where error added to the system produces significantly larger slope values. Furthermore, we can observe that, when error is added to the input, we observe much less significant interaction between the steepness of the given coefficients. We see overlap between the medium and fast ascending curves when noise is added to the input, whereas when noise is added to the system, the difference is much clearer. These results will be utilized for the more detailed analysis as we believe the results are more reflective of the relationships we are trying to understand.

4.2 Sample Size:

Quantifying the relationship between sample size and curve steepness is important because sample size is sometimes controllable in experimentation. Understanding optimum sample sizing and how it may influence the results observed is very useful information for our conclusion.

In this experiment, sample size has a clear and direct influence on the logistic regression fit. This is no surprise and the influence it has on the steepness of the S-Curve is significant. The higher the sample size, the more consistent the simulation and the results. As you can see in the following Figure 11, as sample size increases, we see much tighter groupings of data and higher values of slope.

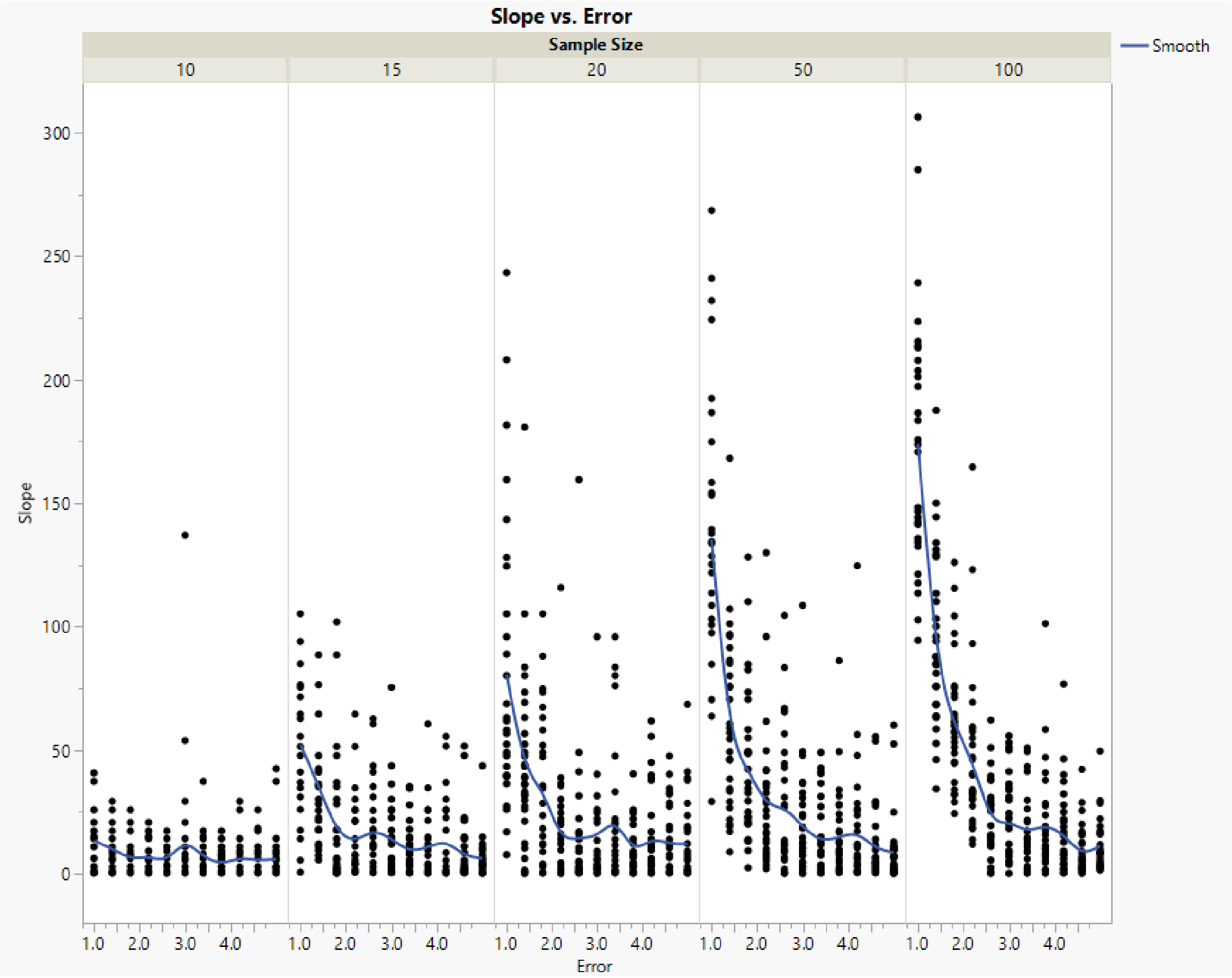


Figure 11: Medium Ascending**,** Error Structure 1 Slopes based on Standard Error and Sample Size.

There appeared to be a convergence in results with a sample size of 50 or greater. We observed almost no discernable difference between the results of simulations with sample size of 50 and 100, but we did notice a difference when the sample size was lower. To confirm this, we analyzed sample sizes between 10 and 20 and the results show that at low sample sizes, the slope of the curve reaches its minimum value much faster. We specifically observed the lower sample sizes for Error Structure 1 of the O-ring data, due to the issues we previously found with Error Structure 2 and low sample sizes.

While these conclusions were useful, none of them were particularly surprising as it is well known that the standard error of the fit will shrink as sample size increases. With a smaller standard error of fit, we will have a tighter confidence band, meaning the points used to calculate our slope will fall much closer to the steepest point of the S-Curve, giving us a greater slope value.

An important characteristic of this result is that the plots appear to follow an exponential decay. With noise added to the input, we observe the medium-ascending coefficients for samples of 50 and 100. Figure 12 shows the results of this comparison, which illustrates little difference between the sample size of 50 and the sample size of 100. These figures indicate that at larger sample sizes exceeding 20 samples, the results become consistent. We would not recommend applying this analysis to experiments where less than 20 samples will be recorded as the results will be too unreliable to draw accurate conclusions.

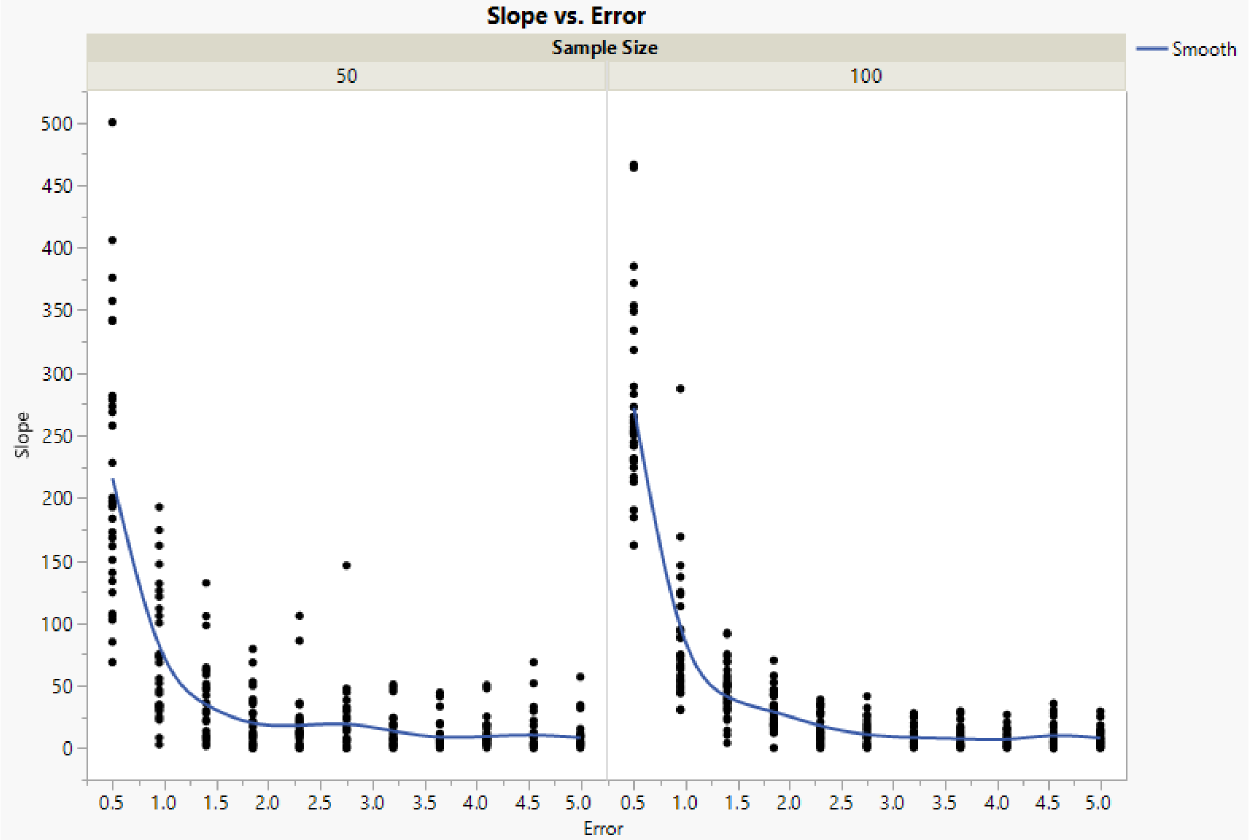


Figure 12: Fast Ascending, Error Structure 2 Slope vs. Error at high sample sizes

4.3 Standard error of simulation

The next essential factor to observe when analyzing the grey area slope is the standard error incorporated into the Monte Carlo simulation. It is important to note that since we are testing multiple error structures in our simulation that the standard error values differ between each structure. To keep the results consistent in our comparisons, we added a scaling factor to the standard error, so that the values are represented as a range of values from -1 to 1. The values at -1 represent the lowest values of error, whereas the values at 1 represent the largest error values.

As the simulation standard error increased, the slope of the S- Curve decreased in all cases. This is reflective of the increased overlap in simulated binary data, leading to a much more gradual S-curve. If the S-curve of our logistic regression model is more gradual than steep, the slope of the grey will be much smaller and in some cases, negative.

In the original test results outline by Table 4, we were observing Error Structure 2 under a much wider range of error than Error Structure 1. Initially, since the slope decrease based on error was slower in Error structure 2, we therefore had to add greater error to see a similar result. If we observe the Medium Ascending coefficients output for Error Structure 2, shown in figure 13, we can see how the slope decreases due to increased error. What the data does suggest is that once error exceeds a certain level, it will tail off and gradually approach 0, asymptotically. This occurs when the mean of the standard error exceeds approximately 1.5. This indicates that if an experiment contains extreme levels of variability, we can expect to see no relationship between the tested variables.

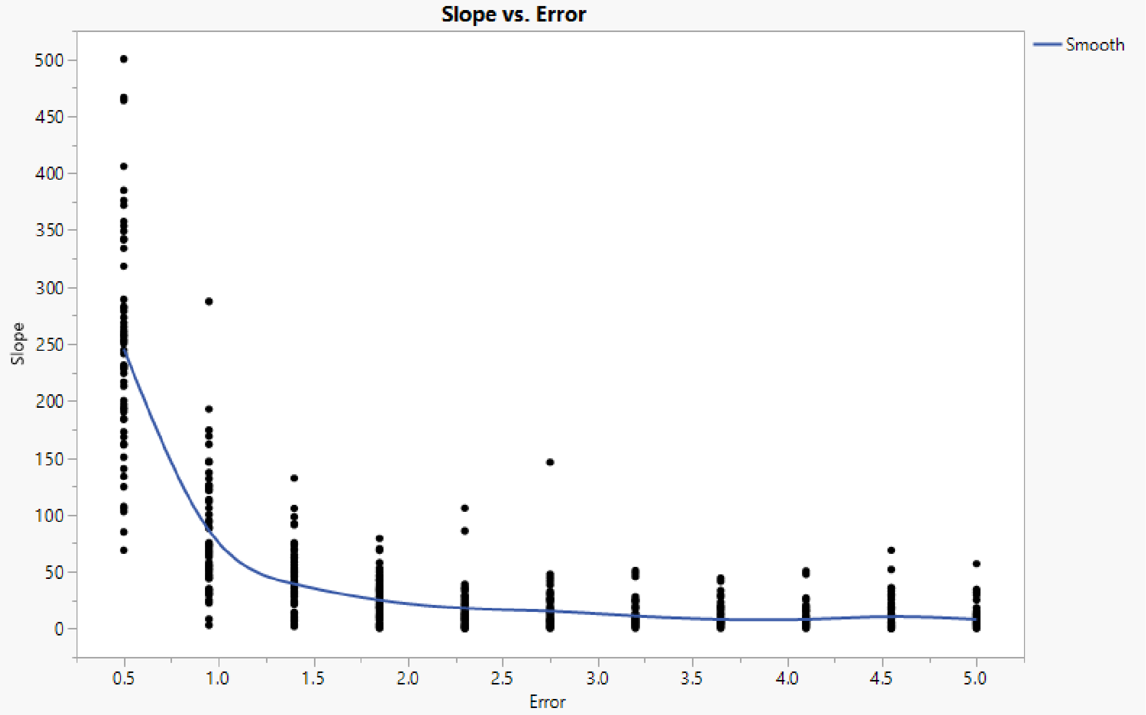


Figure 13: A graph of Slope vs. Error, given a Medium Ascending curve, Error Structure 2, and sample sizes of 50 and 100.

4.4 Significance of test

Another motivation for this experiment was to see if we could detect a significant difference between event probabilities of two input values and relate that result to the slope of our grey area. In our initial testing, we observed whether a significant difference was detected between the minimum and the maximum value of the O-ring data set. From there, we considered different outcomes of the tests including the percent of the input range that the grey area covered and the slope of the grey area. Since this was a very specific application, we determined the results would be more general if they were observed for the 3 different levels of steepness discussed earlier.

For data sets that did not have a defined population, such as the varied coefficients outlined by Kerns (2016), we used the input associated with the 10% probability of success and the input associated with the 90% probability of success. These are values that are often observed in logistic regression analysis with respect to threshold decision making, and it would be important to know if we may discern these values from each other through our simulation.

From initial observations, it was clear that when a grey area included a substantial portion of the input range between the minimum and maximum values in question, it was unlikely that a significant difference would be detected. Therefore, the lower our slope value, the less likely we were to see significant differences between the extremes. For our example, we consider the cases of the fast and slow ascending, at a Sample Size of 100, from our secondary tests with smaller overall standard error.

What we observe is that when noise is added to the input, we see far fewer tests with a significant difference between the extreme points. This result is surprising as it did not appear that the penalty for increased error was so dramatic. At moderate levels of error for noise added to the system, we still see almost all tests show significant difference between extremes. This tells us that very few tests conducted with high levels of noise in the input will reveal stable evidence to make any sort of accurate prediction. It implies the opposite for when noise is added to the system, as only in cases of extreme error are the results not showing a clear detectable difference.

It should be noted that the effect of sample size was exactly as expected and that as sample size increased, we saw a clear increase in the number of significant differences detected. This is of no surprise as when we increase sample size we will decrease the error of the fit, giving us much tighter confidence bands even at extreme points. As a result, this means that when there is overall system variability, significance can be observed even when variability is high. Alternatively, when variability exceeds reasonable levels when added to the input, the results become much less significant.

4.5 O-Ring Analysis

To test the results of this research, we applied our study to the O- Ring Data that was previously discussed. Table 6 displays the results of testing the O-Ring Coefficients at the varied test levels to determine the slope of the middle quantile. These results were tested with moderate standard error values. We observe similar trends to that of the slow ascending curve, which is to be expected given the similarity of the β_1 coefficient.

Table 6: Table summarizing the average slope values that resulted from testing the O-Ring coefficients.

Slope of Middle Quantile							
S-Curve Increase	Sample Size	Error Structure 1			Error Structure 2		
		σ_{Low}	σ_{Med}	σ_{High}	σ_{Low}	σ_{Med}	σ_{High}
O-Ring [-15, .23]	20	2.07	0.72	0.26	2.41	1.64	1.09
	50	8.14	1.16	0.35	10.73	4.72	2.05
	100	12.13	1.28	0.52	20.82	4.80	2.90

Table 7 outlines the percent of the input range that is taken up by the middle quantile that is established by the grey area. The results shown in Table 7 correspond to the experiments run in Table 6.

Table 7: Percent of the initial input range that the middle quantile extends.

Percent of Input Range of Middle Quantile							
S-Curve Increase	Sample Size	Error Structure 1			Error Structure 2		
		σ_{Low}	σ_{Med}	σ_{High}	σ_{Low}	σ_{Med}	σ_{High}
O-Ring [-15, .23]	20	29.08	55.98	123.29	27.24	32.29	39.98
	50	13.65	29.91	239.79	12.48	16.93	22.88
	100	9.62	23.89	44.98	8.10	11.97	16.06

What we observe in this case is how we could apply this thesis to real world applications. Given the O-Ring coefficients, at a sample size of 50, with a low amount of noise added to the whole system, we can anticipate an approximate slope of 8.14 in the middle quantile. In this case, the middle quantile extends across 13.65% of the initial input range. Alternatively, when error increases to a higher level, given the same remaining characteristics, we observe an average slope of 0.35 in the middle quantile that extends across 239.79% of the initial input range.

These slope values tell us that given those model characteristics, in the middle quantile we can anticipate an increase of 8.14% probability of success for every 1 degree increase in temperature on the day of launch. As we increase error, that percent increase drops to 0.35% increase in probability for every 1 degree increase in temperature. The percent of the input range tells us the relative amount of our initial input range that we can expect this relationship to apply.

With these simulations, we can gather relative approximations of anticipated changes in slope and percent of input range, given model characteristics. This allows the user to observe the influence of changing model parameters to account for expected situations.

4.6 Percent of Input Range

The simulation process begins by developing an initial input range based on the model coefficients that spans from the 10% threshold to the 90% threshold. When the simulation analyzes the middle quantile, it records the minimum and maximum values of the range used to calculate the slope and compares them to this initial input range. Given that this middle quantile represents our region of uncertainty, a smaller percentage indicates lower variability of the simulation.

Figure 14 and 15 shows how these percentages vary between error structures and based on the coefficients. Both of these are from our results of low error and sample sizes of 50 and 100 compiled together.

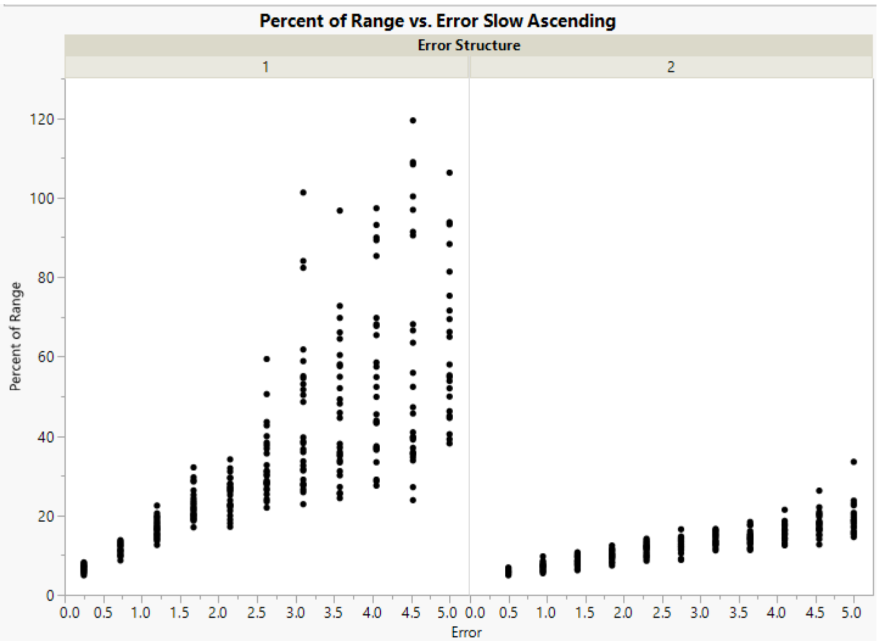


Figure 14: Percent of input range with a slow ascending curve for both Error Structures.

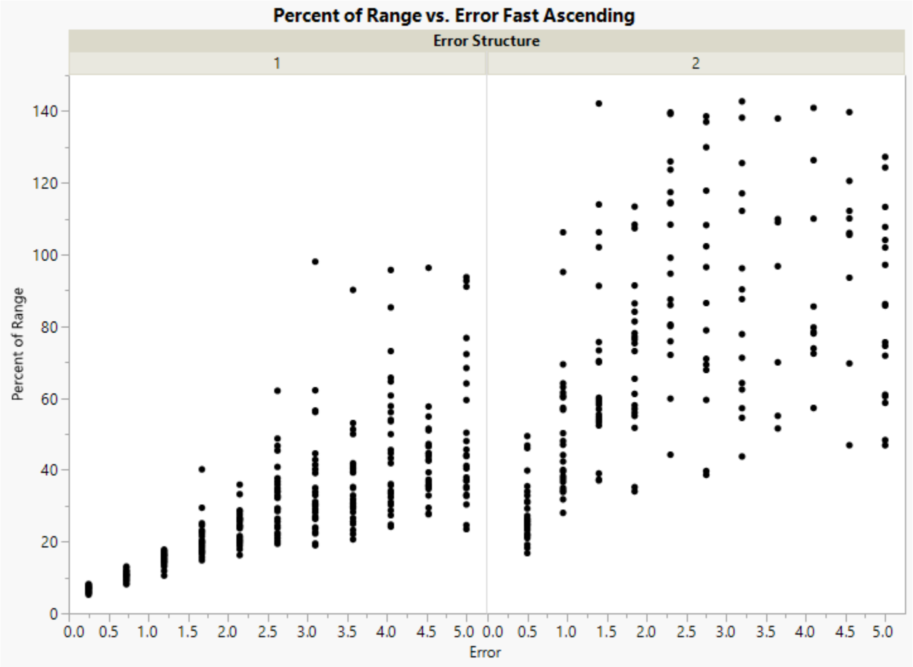


Figure 15: Percent of input range with a fast ascending curve for both Error Structure.

These figures indicate that Error structure 2, or noise to the input, shows much more variability given faster ascending curves, whereas Error Structure 1, or noise to the whole system, is consistent regardless of the steepness of the S-curve. Both situations of noise show that as noise increases, the percent of the input range will also rise.

A likely explanation for the increase in variability with Error Structure 2 is due to the function shown in Equation 7 that shows how this error is added. The level of noise is multiplied by the β_1 coefficient. Thus, when β_1 is greater, the noise increase will have a greater effect than in Error Structure 1. The opposite will occur given smaller β_1 values.

4.7 Linear Regression Analysis of S-Curve Slope

One of the primary objectives of this analysis was to understand if we can quantify the value of the S-curve slope as a function of the data characteristics and the simulation levels. The primary variable analyzed with respect to the S-curve slope will be the standard error of the simulation. This determines how much noise will be added to the original data fit and we anticipate it to have a large influence on the response.

The additional variables in question for this analysis include the sample size of the simulated data, the beta coefficients for the population data, and the error structure used in the simulation. We believe that with these variables known, we can develop an accurate prediction of the approximate steepness of the S-curve response curve that will be produced because of a conducted test. Furthermore, using the developed relationship between the steepness of the curve and the significance analysis at the extreme values, we can conclude based on the model characteristics, whether it is reasonable to expect a statistical significance to be observed.

A simple linear regression applied to each of the three Kerns (2016) coefficients we utilized to see if we could accurately predict the slope value based on our parameters. For this we used a constant sample size of 100 and used the standard error as the only variable in question. It should be noted that there were violations to some of the residual assumptions for the slow and medium ascending slopes when analyzing Error Structure 1.

It was clear from our initial analysis that there appeared to be a logarithmic relationship between error and slope. Given the logistic regression equation involves raising e to the power of the coefficients, this is not very surprising. This does allow us to develop a relative linear relationship after a transformation and we can see unique trends for each level of analysis. Another interesting factor here is that each error structure showed a unique set of coefficients in their linear relationship, while all holding true to the logarithmic fit. This is a very interesting point, as they all showed similar decay in the slope, the rate of the decay varied greatly based on Error Structure and on the coefficients used.

With noise added to the system, the fast ascending showed the greatest rate of decline in slope value, whereas the slow ascending showed the lowest rate. This leads us to believe that if given coefficients that result in a Steep S-Curve, we can anticipate that as error in simulation increases, slope will decline more rapidly. When noise is added to the input, there is no such clear trend. The medium increasing S-Curve showed the greatest decline in slope as error increased and the fast increasing showed the slowest decline. These results are interesting to us as we cannot find a cumulative function to develop the anticipated slope for all models. There is too much dependence on the factors of the Coefficients and the Error Structure.

5. Discussion and Future Work

The coefficients utilized for slow, medium, and fast ascending S-Curves were shown to have relationships we anticipated. The slower the S-Curve ascends, the less steep the slope. Alternatively, with a fast-ascending curve we see much higher slope values. This is applicable based on the scale of the input variables on an experiment. It is easy to determine approximate steepness of an S-curve based on its coefficients and therefore it can be compared to one of our sets of results.

From this coefficient analysis, we can now anticipate relative slope values. If a $\hat{\beta}_1$ is a small value, between 0 and 1, we can expect a slow ascending slope a relatively small slope value in the middle quantile. As $\hat{\beta}_1$ increases, such as values between 1 and 3, we expect a moderate ascending S-Curve and larger slope values in the middle quantile. As $\hat{\beta}_1$ begins to exceed 3, it starts to ascend at a much steeper rate and the slope values will be greater.

The trend indicated that at low error and larger sample sizes, the slope values grew close to the coefficients often exceeding them. As error was higher or sample size was lower, these slope values saw a drop off. This indicates that if we see slope values dramatically smaller than our $\hat{\beta}_1$ coefficient, the slope of the S-Curve is smaller than we should anticipate. That may indicate high error or low sample sizes. Alternatively, if the slope exceeds or approaches $\hat{\beta}_1$, then we can anticipate that we are close to the ideal scenario for that experiment.

When developing this model for analysis, a major point of interest in our simulation was the method of data incorporation. The data structures outlined earlier were both tested at similar experimental levels to observe if the results of the simulations showed any significant difference between the two. Both error structures represent a different scenario in which error may be present in data collection. Understanding where data is most likely to occur will allow this model to provide more specific and useful information to the user.

What we can take away from this analysis is that the slope of the logistic regression curve can be studied to gain a better understanding of how a given experiment will respond to environmental changes. We can use the web application to conduct predictive experiments that allow us to observe the change of the response given increasing error, sample size, and/or changing coefficients. These results can be applied to make generalizations about experiments that utilize coefficients comparable to the set we analyzed. This thesis has helped us to gain a better understanding of the logistic regression model and how it may be applied to applications with uncertain circumstances.

In future work, it may be beneficial to apply this analysis to a wider range of coefficients and to see if that has a direct effect on the rate of change of the slope. It may be valuable to keep the intercept values consistent as well. Additional methods of simulation could provide interesting results that could be compared to this thesis. Along with simulation methods, any additional random error generation methods may provide interesting results. In all, it is likely very challenging to create a model that can accurately predict the slope values of a curve based on the simulation settings due to the wide range of variables that have produced unique results.

Works Cited

Bender, R., & Grouven, U. (1996). Logistic regression models used in medical research are poorly presented. *BMJ : British Medical Journal*, *313*(7057), 628.

Hosmer, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). Wiley Series in Probability and Statistics : Applied Logistic Regression (3rd Edition). New York, NY, USA: John Wiley & Sons.

Hurwitz, A. and Remund, T. (2014). A Large-Sample Confidence Interval for the Inverse Prediction of Quantile Differences in Logistic Regression for Two Independent Tests. *Quality Engineering*, 460-466.

Kerns, L. (2016). Construction of simultaneous confidence bands for multiple logistic regression models over restricted regions. *Statistics.* 50(6), 1332-1345. doi: 10.1080/02331888.2016.1230616

Kist, M. J., and Silvestrini, R. T. (2016). Incorporating Confidence Intervals on the Decision Threshold in Logistic Regression. *Qual. Reliab. Engng. Int.*, 32(5), 1769–1784. doi: 10.1002/qre.1912.

Lu, W., Ramsay, J., and Bailey, J. (2000). Reliability of Pharmacodynamic Analysis by Logistic Regression. *Anesthesiology*, 1255-1262.

Maranzano, Coire J. and Roman Krzysztofowicz. (2008). "Bayesian Reanalysis of the Challenger O-Ring Data." *Risk Analysis: An International Journal*, vol. 28, no. 4, pp. 1053-1067. EBSCO*host*, doi:10.1111/j.1539-6924.2008.01081.x.

Stoltzfus, J. C., (2011). Logistic Regression: A Brief Primer. *Academic Emergency Medicine*, 18(10), 1099-1104. doi: 10.1111/j.1553-2712.2011.01185.x

Tan, K., Sulke, N., Taub, N., Watts, E., Karani, S., & Sowton, E. (1993). Determinants of success of coronary angioplasty in patients with a chronic total occlusion: A multiple logistic regression model to improve selection of patients. *Heart*, 126-131.

Van Ewijk, P., & Hoekstra, J. (1994). Curvature Measures and Confidence Intervals for the Linear Logistic Model. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 43*(3), 477-487. doi:10.2307/2986272

Appendix

1.0 Shiny Application Code

User Interface:

```
# Interactive Logistic Regression Model
library(shiny)
# Define UI for application that draws a histogram
shinyUI(fluidPage(
  # Application title
  titlePanel("Logistic Regression Grey Area Model"),
  sidebarLayout(
    sidebarPanel(
       #sliderInput("Threshold",
        #           "Threshold Value:",
         #          min = .30,
          #         max = .70,
           #        value = .50),
       sliderInput("Sample Size",
                   "Sample Size:",
                   min = 10,
                   max = 100,
                   value = 50),
      # numericInput("SigVal1",
       #              "Upper Significant Value:",
        #             value = 80),
       #numericInput("SigVal2",
        #             "Lower Significant Value:",
         #            value = 50),
      radioButtons("Error Structure", "Select the desired random error structure",
                   choices = list("Option 1" = 1, "Option 2" = 2)),
      numericInput("Beta0",
                   HTML(paste("&beta;",tags$sub(0), ":", sep = "")),
                    value = -15.0429),
      numericInput("Beta1",
                   HTML(paste("&beta;",tags$sub(1), ":", sep = "")),
                   value = .2321627),
      numericInput("Error min",
                    "Minimum Standard deviation of error:",
                    min = .5,
                    value = 1),
      numericInput("Error max",
                    "Maximum Standard deviation of error:",
                    min = .5,
                    value = 3),
     # numericInput("Levels",
      #             "Number of levels in the error:",
       #            min = 1,
        #           value = 11),
      #numericInput("Trials",
       #            "Number of Trials:",
        #           min = 1,
         #          value = 30),
      submitButton(text = "Run Simulation"),
      downloadButton('downloadData', 'Download Raw Slope Data'),
      downloadButton('downloadSlope', 'Download Averaged Slope Data')
      ),
    mainPanel(
      plotOutput("GreyAreaModel"),
      tabsetPanel(
      tabPanel("Application Instructions",
               titlePanel("Welcome!"),
               h4("Thank you for taking the time to use our interactive Logistic Regression
Grey Area analysis application.
```

```
		Below you'll find some basic instructions as to how to utilize the application.
		This application was developed by Michael J Kist with the assitance of Dr.
Rachel Silvistrini."),
	titlePanel("User Interface:"),
	titlePanel("Sample Size"),
	h4("Each Simulation has data randomly generated based on your choice of
sample size.
		To change the sample size, simply move the slide panel to the desired
value."),
	titlePanel("Error Structure"),
	h4("We are experimenting with multiple different error structures that are
applied to our simulated data.
		The first error structure adds noise to the entire system by summin it to the
intercept coefficient.
		The second error structure adds error to the input variable being tested, to test
the accuracy of the measurment specifically.
		To choose your error structure simply select the given bubble associated with
the structure you wish to test.
		"),
	titlePanel("Logistic Regression Coefficients"),
	h4("This application allows you to test specific data sets by providing input
options for the logistic regression coefficients.
		In order to test your specific example, adjust the coefficients to match the
Logistic Regression results of your historical data.
		The default coefficients represent the results of the O Ring failure data leading
up to the 1986 Challenger Launch in which a critical failure occurred.
		"),
	titlePanel("Range of the Simulation Standard Error"),
	h4("In order to produce unique results, the model produces a data set based on
the known regression results and a specific level of noise.
```

```
            These values represent the amount of noise added to the simulation to produce
our new data.  The standard error will be 11 values, evenly spread between the minimum
and maximum value input.
            ")
    ),
     tabPanel("Confusion Martices",
           titlePanel("Prediction Matrix"),
           tableOutput("Prediction Matrix"),
           titlePanel("Grey Area Prediction Matrix"),
           tableOutput("Grey Prediction Matrix")
    ),
    tabPanel("Grey Area Size",
          h3(textOutput("Grey Area Size")),
          h3(textOutput("Grey Area Percent")),
          h3(textOutput("Prob Area Size")),
          h3(textOutput("Grey Area Slope"))
    ),
    tabPanel("Data Coefficients",
          h3(HTML(paste(
             "<math xmlns='http://www.w3.org/1998/Math/MathML'>
                  <mover accent='true'>
                    <mi>&beta;</mi>
                     <mo>&Hat;</mo>
                 </math>"
          ,tags$sub(0), ": ", textOutput("Coefficient 0", inline = TRUE), sep = ""))),
          withMathJax(textOutput('')),
          h3(HTML(paste(
            "<math xmlns='http://www.w3.org/1998/Math/MathML'>
                  <mover accent='true'>
                     <mi>&beta;</mi>
                     <mo>&Hat;</mo>
```

```
                          </math>"
               ,tags$sub(1), ": ",textOutput("Coefficient 1", inline = TRUE), sep = "")))
      ),
      tabPanel("Slope Data",
              titlePanel("Slope Means"),
              tableOutput("SlopeSummary"),
              titlePanel("Slope Data"),
              tableOutput("SlopeTable")
      ),
      tabPanel("Slope Trend",
               plotOutput("SlopeTrend")
      ))))))
```

Server:

```
shinyServer(function(input, output, clientData, session){
  output$GreyAreaModel <- renderPlot({
    ###Input Values###
    Beta0 <- input$Beta0
    Beta1 <- input$Beta1
    Limits <- c(round((log(.9/(.1))+Beta0)/(-Beta1), digits = 0),
round((log(.1/(.9))+Beta0)/(-Beta1), digits = 0)) ###Calculated upper and lower range
value
    lowA <- min(Limits)
    highA <- max(Limits)
    t <- .5
    CI <- 1.96
    n <- input$`Sample Size` #Record the desired sample size based on user input
    ### Initialize Data Collection
    PredMat <- data.frame("Predicted 1" = c(0,0), "Predicted 0"= c(0,0),
                          row.names = c("Actual 1", "Actual 0"))  #Set up the prediction matrix
    GreyMat <- data.frame("Predicted 1" = c(0,0), "Predicted 0"= c(0,0),
                          row.names = c("Actual 1", "Actual 0"))  #Set up the prediction matrix
    SlopeTable <- data.frame()
    slopeSummary <- data.frame()
    SigTable <- data.frame()
    #The following function generates a new set of data based on the input parameters
    Emin <- input$`Error min`
    Emax <- input$`Error max`
    Levels <- 11
    TestValues <- seq(from = Emin, to = Emax, length.out = Levels)
    Trials <- 30
    for(i in 1:length(TestValues)){
      SlopeSum <- 0
      PercentRangeSum <- 0
      SigSum <- 0
      XMinSum <- 0
      XMaxSum <- 0
      for(w in 1:Trials){
       GenerateData <- function(len = n, CurrentSigma = TestValues[i]) {
        range <- seq(from = lowA, to = highA, length.out = len)  #Apply a range of values
based on the data
        if(input$`Error Structure` == 1 ){ ##Option 1 error (Standard)
          z = Beta0+Beta1*(range)+rnorm(n, mean = 0, sd = CurrentSigma)
          }
        if(input$`Error Structure` == 2 ){ ##Option 2 error (add to input variable)
          z = Beta0+Beta1*(range+rnorm(n, mean = 0, sd = CurrentSigma))
          }
        pr = 1 /(1+exp(-z)) ##calculate the probabilities
        y = round(pr) ##Determine the decision based on the threshold of .5
        ynew = y[order(y)]
        if(sum(ynew == y) > n - 3){
          GenerateData()
        }else{
          data.frame(range, pr, y)
           }
         }
    CurrentData <- GenerateData()
    glm.fit <- glm(y~range, data = CurrentData, family = binomial(link = "logit"))
    LimitsB <- c(round((log(.9/(.1))+glm.fit$coeff[1])/(-glm.fit$coeff[2]), digits = 0),
round((log(.1/(.9))+glm.fit$coeff[1])/(-glm.fit$coeff[2]), digits = 0)) ###Calculated upper
and lower range value
     lowB <- min(LimitsB)
    highB <- max(LimitsB)
    std.erLow <- function(X){
      pred <- predict(glm.fit, newdata = data.frame(range = X), se.fit = TRUE)
      exp(pred$fit-CI*pred$se.fit)/(1+exp(pred$fit-CI*pred$se.fit))
    }
    std.erHigh <- function(X){
      pred <- predict(glm.fit, newdata = data.frame(range = X), se.fit = TRUE)
      exp(pred$fit+CI*pred$se.fit)/(1+exp(pred$fit+CI*pred$se.fit))
    }
    #range <- seq(from = lowB, to = highB, length.out = n)  #Apply a range of values
based on the data
    k <- (log((1-t)/t)+glm.fit$coeff[1])/(-glm.fit$coeff[2])  #Determinine the success
threshold for the x values
    PU <- std.erHigh(k)
    PL <- std.erLow(k)
    GU <- (log(PU/(1-PU))-glm.fit$coeff[1])/glm.fit$coeff[2]  #Functions to determine the
"grey area"
    TGU <- as.numeric(round(GU, digits = 2))
    GL <- (log(PL/(1-PL))-glm.fit$coeff[1])/glm.fit$coeff[2]
    TGL <- as.numeric(round(GL, digits = 2))
    PS <- PU - PL
    GS <- GU - GL
    XMin <- GL
    XMax <- GU
    ## Intersections for limiting Grey Area Graphically
    UCI <- std.erHigh(TGL)
    LCI <- std.erLow(TGU)
    Slope <- 100*(PS/(GS*GS))
    Error <- TestValues[i]
    ScaledError <- (Error - (Emin + Emax)/2)/((Emax-Emin)/2)
    PercentRange <- abs(round(100*(GS/(highA-lowA)), digits = 2))
    Sig1 <-  std.erLow(highA)  #std.erLow(input$SigVal1)
    Sig2 <-  std.erHigh(lowA) #std.erHigh(input$SigVal2)
    if(Sig1 >= Sig2){
     Significance = "Yes"
     Sig <- 1
    }else{
     Significance = "No"
     Sig <- 0
    }
    SlopeTable <- rbind(SlopeTable, data.frame(w, n, input$`Error Structure`, Error, ScaledError, Slope, XMin, XMax, Significance, PercentRange))
    SlopeSum <- SlopeSum + Slope
    PercentRangeSum <- PercentRangeSum + PercentRange
    SigSum <- SigSum + Sig
    XMinSum <- XMinSum + XMin
    XMaxSum <- XMaxSum + XMax
    SigTable <- rbind(SigTable, data.frame(n, input$`Error Structure`, Error, ScaledError, Significance))
    }
    SlopeAvg <- SlopeSum/Trials
    PercentRangeAvg <- PercentRangeSum/Trials
    SigPerc <- SigSum/Trials
    XMinAvg <- XMinSum/Trials
    XMaxAvg <- XMaxSum/Trials
    slopeSummary <- rbind(slopeSummary, data.frame(n, input$`Error Structure`, Error, ScaledError,SlopeAvg,XMinAvg, XMaxAvg, SigPerc,PercentRangeAvg))
    }
  with(CurrentData, plot(y~range, type="p", xlim = c(lowB,highB), ylim=c(0, 1),
                         main = "Simulated Logistic Regression",
                         ylab="Probability of Success",
                         xlab="Input Variable"))  #Plot the data points, label the graph
    curve(predict(glm.fit,data.frame(range=x),type="resp"),add=TRUE) # draws a curve based on prediction from logistic regression model
    curve(std.erLow, lowB, highB, n=n, add = TRUE, lty = 2, col = "blue")
    curve(std.erHigh, lowB, highB, n=n, add = TRUE, lty = 2, col = "blue")
    segments(GU,PU,k,PU)
    segments(GU,LCI,GU,PU)
    segments(GL,PL,GL,UCI)
    segments(GL,PL,k,PL)
    segments(GL,t,GU,t)
    segments(k,PU,k,PL)
    SlopeTable
    slopeSummary
    colnames(SlopeTable) <- c("Trial", "Sample Size", "Error Structure", "Error", "Scaled Error", "Slope", "Xmin", "XMax", "Significance", "Percent of Range")
    colnames(slopeSummary) <- c("SampleSize", "Error Structure","Error", "Scaled Error", "Average Slope","Average Xmin", "Average XMax", "Average Significance", "Average Percent")
    A0P0Data <- subset(CurrentData, range < GL & y == 0)
    A1P0Data <- subset(CurrentData, range < GL & y == 1)
    A0P0GData <- subset(CurrentData, range <= k & range >= GL & y == 0)
    A1P0GData <- subset(CurrentData, range < k & range >= GL & y == 1)
```

```
     A0P1GData <- subset(CurrentData, range > k & range <= GU & y == 0)
    A1P1GData <- subset(CurrentData, range >= k & range <= GU & y == 1)
    A1P1Data <-subset(CurrentData, range > GU & y == 1)
    A0P1Data <-subset(CurrentData, range > GU & y == 0)
    A <- nrow(A0P0Data)/n
    B <- nrow(A1P1Data)/n
    C <- A + B
    D <- nrow(A0P1Data)/n
    E <- nrow(A1P0Data)/n
    G <- D + E
    H <- nrow(A0P0GData)/n
    I <- nrow(A1P1GData)/n
    J <- H + I
    K <- nrow(A0P1GData)/n
    L <- nrow(A1P0GData)/n
    M <- K + L
    PredMat[1,1] <- B
    PredMat[2,2] <- A
    PredMat[1,2] <- E
    PredMat[2,1] <- D
    GreyMat[1,1] <- I
    GreyMat[2,2] <- H
    GreyMat[1,2] <- L
    GreyMat[2,1] <- K
    output$SlopeTrend <- renderPlot(
      plot(slopeSummary$'Average Slope' ~ slopeSummary$'Scaled Error', type = "b", xlab
= "Scaled Error", ylab = "Average Slope", main = "Slope Trend")
    )
       output$downloadData <- downloadHandler(
       filename = function(){
         paste("Raw Slope data-", Sys.Date(), ".csv", sep = "")
         },
       content = function(file){
         write.csv(SlopeTable, file, row.names = FALSE)
         }
     )
     output$downloadSlope <- downloadHandler(
       filename = function(){
         paste("Avg Slope data-", Sys.Date(), ".csv", sep = "")
       },
       content = function(file){
         write.csv(slopeSummary, file, row.names = FALSE)
       }
     )
    output$`Grey Area Size` <- renderText({
      paste("Grey area input Variable length: ", round(GS, digits = 2), sep = " ")
    })
    output$`Grey Area Percent` <- renderText({
      paste("Percentage of input range: ", round(100*(GS/(highA-lowA)), digits = 2), sep =
" ")
    })
    output$`Grey Area Slope` <- renderText({
      paste("Grey Area Slope: ", round(Slope, digits = 4), sep = " ")
    })
    output$`Prob Area Size` <- renderText({
      paste("Grey area output Probability height: ", round(PS, digits = 4), sep = " ")
    })
    output$`Coefficient 0` <- renderText({
      round(glm.fit$coeff[1], digits = 4)
    })
    output$`Coefficient 1` <- renderText({
      round(glm.fit$coeff[2], digits = 4)
```

```
  })
output$`Prediction Matrix` <- renderTable({
     PredMat
     },
     bordered = TRUE, rownames = TRUE
  )
  output$`Grey Prediction Matrix` <- renderTable({
     GreyMat
  },
  bordered = TRUE, rownames = TRUE
  )
  output$SlopeSummary <- renderTable({
     slopeSummary
  },
  bordered = TRUE, rownames = FALSE
  )
  output$SlopeTable <- renderTable({
     SlopeTable
  },
  bordered = TRUE, rownames = FALSE
  )
  })
})
```