2009

Deriving Optimal Composite Scores: Relating Observational/Longitudinal Data with a Primary Endpoint

Rhonda Ellis
Virginia Commonwealth University

DERIVING OPTIMAL COMPOSITE SCORES: RELATING OBSERVATIONAL/LONGITUDINAL DATA WITH A PRIMARY ENDPOINT

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

RHONDA DENISE ELLIS
B.S., Hampton University, 2001
M.S., Hampton University, 2003

Director: Dr. Chris Gennings
Professor
Department of Biostatistics

Virginia Commonwealth University
Richmond, Virginia
September 2009

Acknowledgement

"If I have seen further it is by standing on the shoulders of giants."

Isaac Newton

Table of Contents

List of Tables

Table **Page**

List of Figures

Figure **Page**

Abstract

Deriving Optimal Composite Scores:
Relating Observational/Longitudinal Data with a Primary Endpoint

By Rhonda Ellis

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2009

Major Director: Dr. Chris Gennings
Professor, Department of Biostatistics

In numerous clinical/experimental studies, multiple endpoints are measured on each subject. It is often not clear which of these endpoints should be designated as of primary importance. The desirability function approach is a way of combining multiple responses into a single unitless composite score. The response variables may include multiple types of data: binary, ordinal, count, interval data. Each response variable is transformed to a 0 to1 unitless scale with zero representing a completely undesirable response and one representing the ideal value. In desirability function methodology, weights on individual components can be incorporated to allow different levels of importance to be assigned to different outcomes. The assignment of the weight values are subjective and based on individual or group expert opinion. In this dissertation, it is

our goal to find the weights or response variable transformations that optimize an external empirical objective criterion. For example, we find the optimal weights/transformations that minimize the generalized variance of a prediction regression model relating the score and response of an external variable in pre-clinical and clinical data. For application of the weighting/transformation scheme, initial weighting or transformation values must be obtained then calculation of the corresponding value of the composite score follows. Based on the selected empirical model for the analyses, parameter estimates are found using the usual iterative algorithms (e.g., Gauss Newton). A direct search algorithm (e.g., the Nelder-Mead simplex algorithm) is then used for the minimization of a given objective criterion i.e. generalized variance. The finding of optimal weights/transformations can also be viewed as a model building process. Here relative importance levels are given to each variable in the score and less important variables are minimized and essentially eliminated.

CHAPTER 1

Introduction and Prospectus

1.1 Introduction

The measurement of multiple outcome variables is common in numerous clinical and experimental studies. A single primary outcome of interest in many cases can not be specified or it is just not clear which endpoint should be designated as so. In the quality engineering community, it is often the case that many properties are not only of interest but need to be balanced. In Kim et al (2000), various mechanical properties of steel are of interest. In an example by Khuri and Conlon (1981) they describe a study in which the effects of several variables on the foaming properties of whey protein concentrates were investigated. In this case, four dependent variables were selected for maximization. From an experimental point of view, investigators felt that all four variables were relevant.

In the toxicology literature, many of the dose-response studies are designed to measure multiple outcomes on each of the experimental subjects. In the examination of neurotoxicity for example, since the range of behavioral functions that may be affected by the exposure to the toxic agent is wide, investigators typically use sets of test. The sets are known as test batteries and each battery could include as many as 30 tests (Moser 1997). Comprehensive assessments are done on agents that pose high risk to public health and national security. Animal models are used to better understand these agents. The consensus cognitive battery developed by the National Institute of Mental Health's

(NIMH's) Measurement and Treatment Research to Improve Cognition in Schizophrenia (MATRICS) initiative includes 10 independently developed tests that are recommended as the standard battery for clinical trials of cognition-enhancing interventions for schizophrenia (Kern et al. 2008).

Many of these evaluations involve multivariate data repeated over time. When there are several outcomes of interest, these variables can be aggregated into a single composite score. With the reduction of dimensionality, standard univariate analyses can be preformed, e.g., testing for the significance of a slope parameter in a regression model.

The primary objective of this research has been the development of methodology for creating a composite score that combines a set of multiple response variables that may include different data types (binary, count, ordinal etc.) and optimally linking this set to an external outcome variable through an objective function of interest. The composite score combines response variables with different data types, using the desirability-function approach, by transforming each variable to the unit interval where a value of 1 is the absolute best case and a value of 0 indicates an absolute worse case. The values of the individual components of the composite score indicate the relative importance of the variables. Our goal is to link a set of response variables, of possible mixed data types, through a single aggregate to an external outcome variable such as death or disease progression. The result is a simple composite score that can be used to track the outcome variable, for example, disease progression. There is value in simplicity and this composite score is visual and very user friendly.

The methodology includes a ‘variable selection’ aspect which determines a subset of variables that are included in the composite score. In addition, the methodology incorporates the subset of variables in the composite score based on their relative importance.

1.2 Prospectus

This dissertation is written in a distinct style. Chapters 3 and 4 are preliminary versions of manuscripts in preparation to be submitted to statistical and medical journals. These chapters are meant to stand alone. For this reason each chapter will contain its own introduction, brief literature review, methods and conclusion. Because of similarities, there may be some overlap in forth coming chapters.

A literature review on multi-response optimization is given in Chapter 2. Here a full review is given on desirability functions and their corresponding shapes. A few examples of how desirability functions have been implemented in various literatures are also shown in Chapter 2. Chapter 3, The Development and Analysis of a Morbidity Score Using Optimal Transformations of Desirability Functions, describes the development and validation of methodology for creating a composite score that combines multiple response variables which is optimally related to an empirical objective function. The methodology includes a ‘variable selection’ aspect which determines a subset of variables that are included in the composite score. In addition, the methodology incorporates the subset of variables in the composite score based on their relative importance. In a pre-clinical example, morbidity evaluations of animals that involve multivariate data including observational, biological and behavioral variables are measured repeatedly. In this research we develop a morbidity composite score where the observational outcomes are synthesized into a single score with validation of the score based on its statistical relationship to instantaneous hazard of death. Optimal weights/transformations of the multiple response variables that comprise the score were determined by using a nonlinear optimization subroutine for parameter estimation for the hazards model embedded within the direct search algorithm controlling the weights/transformations. The objective is to determine the weights/transformations such that a criterion is satisfied (eg. the generalized variance of the Cox regression model is minimized). Several transformations of the score are considered and compared. In addition, we apply the penalized optimality criterion by Parker and Gennings (2008) to improve the practicality of the designs. For Illustration purposes, one transformation method is chosen and the penalized method is implemented.

As a demonstration of how this method can be implemented in a clinical study, Chapter 4 describes the creation and use of a severity index in a pancreatitis study. The objective is to develop a severity index which can be used to track the progress of the disease and to potentially predict a worsening condition using variables describing patient behavior (e.g., smoking, concurrent drinking, age, gender) and physiological measurements from MRI (e.g., side branch size, contour abnormality of the bile duct). Pancreatitis severity in this analysis is defined as patients having any of the following six months from baseline: exocrine failure, diabetes, pseudocyst and bile duct stricture. An ordinal 'response' score was created which counted the number of these conditions for each patient. Of particular interest is to determine how predictive such a composite score can be in predicting the likelihood of a decline in disease status over the next six months.

Chapter 5, Summary and Future work, concludes this dissertation with a summary of our contribution to the development of statistical methodology for the creation of optimal composite scores and a discussion of future extensions and applications.

CHAPTER 2

Multi-response Optimization

2.1 The Multi-response Problem

Suppose we have k response variables $\mathbf{Y} = \left(Y_1, Y_2, ..., Y_k\right)$ that are may be associated by p independent variables $\mathbf{X} = \left(X_1, X_2, ..., X_p\right)$. The multiple response problem in general could be defined as

$$Y_i = f_i\left(X_1, X_2, ..., X_p\right) + \varepsilon_i \quad \text{for} \quad i = 1, 2, ..., k$$

where $\mathbf{f} = \left(f_1, f_2, ..., f_k\right)$ represents the functional relationship between Y_i and $X_1, X_2, ..., X_p$ and the function may differ for each of the Y_i's. In practice, the exact form of the f_i's are generally unknown and are typically estimated using model building techniques, i.e. regression. In the engineering community, it is of interest to be able to select a set of conditions (X's) which will result in a product with a desirable combination of properties (Y's) (Derringer and Suich 1980). A straightforward approach to the multi-response optimization problem would be to superimpose the response contour plots and visually inspect the optimum point. Although this method is simple, it is limited to cases where the responses are few. Derringer and Suich (1980) mention that the problem in using linear programming techniques is that it optimizes one response variable subject to constraints of the other remaining responses. As one product improves, it is normally at the expense of one or multiple properties. Often times investigators want to attain the best balance among several response variables. Xu et al. (2004) state that although the conventional experimental design and model techniques are still useful, the challenge is how to simultaneously determine the optimum factor settings for the multiple responses and attain the overall desired quality. They also mention that with the increasing demand for the attainment of overall quality that a systematic and robust strategy to optimize all responses simultaneously is crucial. Difficulties arise when trying to decide how to average properties measured in different units.

2.2 The Generalized Distance Approach

The concept of addressing the optimization of multiresponse systems using the generalized distance approach was proposed by Khuri and Conlon (1981). The proposed two-step process was to first obtain the individual optima of the k estimated responses over the experimental region. Next, find the combined optimum by minimizing the distance function by measuring the deviation from the ideal optimum. The distance measure can be defined as (Kim and Lin, 2000):

$$\rho\left[\hat{y}(x),\phi\right]=\left[\left(\hat{y}(x)-\phi\right)'\left\{\mathrm{var}\left[\hat{y}(x)\right]\right\}^{-1}\left(\hat{y}(x)-\phi\right)\right]^{\frac{1}{2}} \qquad (2.1)$$

where ϕ is the optimum value of $\hat{y}\ (x)$, the vector of predicted responses at x and $\mathrm{var}\left[\hat{y}(x)\right]$ is the variance-covariance matrix of the predicted response. This approach also takes into consideration the variation caused by the randomness of ϕ by minimizing an upper bound on the distance within the confidence region of ϕ.

The generalized distance approach is limited because it requires that all predicted response functions are identical with respect to the set of input variables and the functional form of these input variables. Also, all responses are assumed to be of the same importance thus no preferences are considered (Xu et al., 2004). A similar method was introduced by Church (1978) where he used the Euclidean distance to measure the deviation from the ideal optimum instead of equation (2.1). Another modification to the Khuri and Conlon method was proposed by Pignatiello (1993) where in this case the expected value of the loss function is:

$$\hat{\mathbf{E}} = \left[\left(\hat{\mathbf{y}}(\mathbf{x}) - \phi\right)' \mathbf{C}\left(\hat{\mathbf{y}}(\mathbf{x}) - \phi\right)\right] + trace\left[\mathbf{C}\left\{\mathrm{var}\left[\hat{\mathbf{y}}(\mathbf{x})\right]\right\}\right] \tag{2.2}$$

where $\mathbf{C}$ is a positive definite matrix of costs. The other terms are as described in (2.1). The first part of the equation $\left[\left(\hat{\mathbf{y}}(\mathbf{x}) - \phi\right)' \mathbf{C}\left(\hat{\mathbf{y}}(\mathbf{x}) - \phi\right)\right]$ represents the penalty added for a response that deviates from the target response. The penalty imposed for the quality of the prediction is represented by the remaining portion of the equation $trace\left[\mathbf{C}\left\{\mathrm{var}\left[\hat{\mathbf{y}}(\mathbf{x})\right]\right\}\right]$. This method takes into account the effect of the predictive ability on the optimal solution however, the difficulty with this method is that the choice of $\mathbf{C}$ is subjective and the computation of the variance-covariance matrix is complicated for practitioners when the response have different model forms (Xu et al., 2004). The two components of equation (2.2) are equally weighted (Xu et al. 2004). That is, the penalty for the deviation $\left[\left(\hat{\mathbf{y}}(\mathbf{x})-\phi\right)'\mathbf{C}\left(\hat{\mathbf{y}}(\mathbf{x})-\phi\right)\right]$ is given the same importance as the penalty for poor quality of the response predictions $trace\left[\mathbf{C}\left\{\mathrm{var}\left[\hat{\mathbf{y}}(\mathbf{x})\right]\right\}\right]$.

2.3 The Desirability Approach

The desirability approach is the most popular method to optimize the multiple quality characteristics problem (Carlyle et al., 2000). Derringer (1994) states that if the properties could be measured on the same scale then one could just take the average of such properties and maximize it. This is the motivating purpose and main idea of the desirability function. In essence, the desirability function condenses a multivariate optimization problem into a univariate one (Derringer and Suich 1980).

Desirability functions, which were introduced by Harrington (1965), transforms each estimated response $\hat{Y}_i$ to a desirability value d_i, $0 \le d_i \le 1$, $i = 1,2,...,k$. Individual d_i's are combined using the geometric mean

$$D = \left(d_1 \times d_2 \times ... \times d_k\right)^{\frac{1}{k}}. \tag{2.3}$$

The single D value gives an overall assessment of the desirability of the combined response levels (Derringer 1994). Desirability, D, will increase as the balance of the properties become more favorable and has the property that if any response renders an unacceptable response ($d_i = 0$) the overall Desirability will be unacceptable, D=0. It is this reason that the geometric mean is preferred over the use of the arithmetic mean. Derringer (1994) introduced a weighted composite desirability described as:

$$D_i = \left(d_1^{w_1} \times d_2^{w_2} \times \ldots \times d_{K}^{w_K}\right)^{1 \Big/ \sum_1^K w_i} . \qquad (2.4)$$

In addition to the desirability curves, each property is associated with a given weight. The weights allow different importance levels to be assigned to different properties. Both the desirability curves and the weight assignments are selected by individual or group judgment and are best done by a consensus (Derringer 1994). It is usually difficult to select the weights that measure the relative importance associated with each objective in the weighted sum method (Xu et al., 2004).

Two cases arise when transforming each response to desirabilities which are the one-sided and two-sided desirability transformations. When Y_i is to be maximized (bigger-is-better; Figure 2.1), that is when d_i increases as Y_i increases Derringer and Suich (1980) employ the transformation that takes the form:

$$d_i = \begin{cases} 0 & Y_i \le Y_{i*} \\ \left[\dfrac{Y_i - Y_{i*}}{Y_i^* - Y_{i*}} \right]^r & Y_{i*} < Y_i < Y_i^* \\ 1 & Y_i \ge Y_i^* \end{cases} \qquad (2.5)$$

where, for response i, Y_{i*} is the minimum acceptable value of Y_i and

Y_i^* gives the highest value of Y_i. However, since this is a one-sided transformation Y_i^* can be thought of as a value such that any value higher would add little to no merit. The variable r is a shape parameter. Note that Y_{i*}, Y_i^* and r are user specified. The shapes in Figure 2.1 demonstrates the flexibility of equation (2.5).

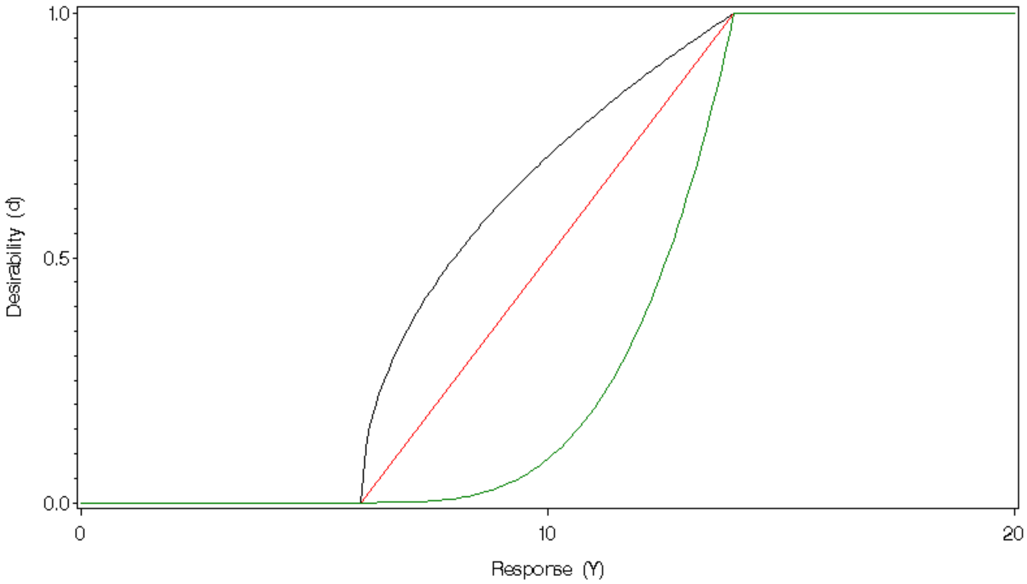


Figure 2.1 Maximization of Response Y (bigger-is-better).

When Y_i is to be minimized (smaller-is-better, Figure 2.2), that is d_i decreases as Y_i increases, the desirability can be expressed as:

$$d_i = \begin{cases} 1 & Y_i \le Y_i^* \\ \left[\dfrac{Y_i - Y_i^*}{Y_{i*} - Y_i^*}\right]^s & Y_i^* < Y_i < Y_i^* \\ 0 & Y_i \ge Y_i^* \end{cases} \tag{2.6}$$

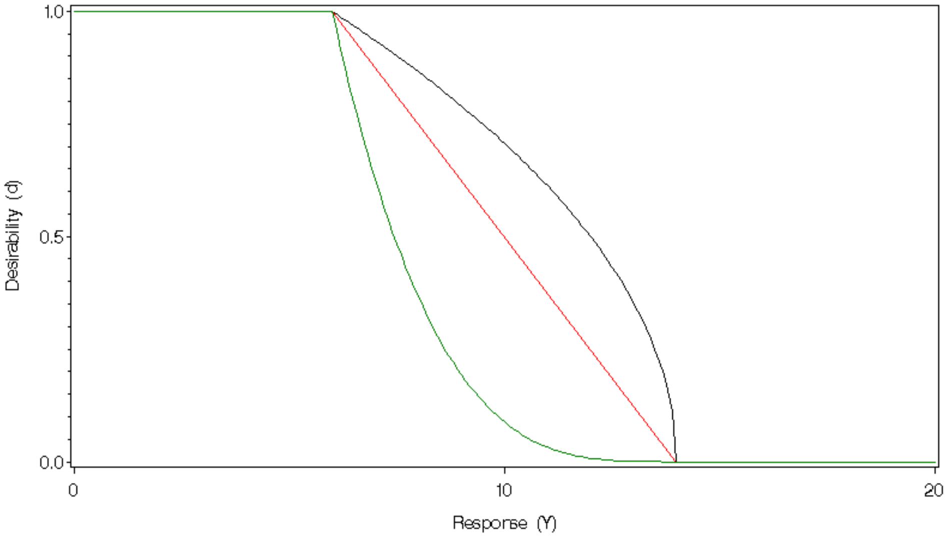


Figure 2.2: Minimization of response Y (smaller-is better).

The two-sided case (target-is-best; Figure 2.3) is considered when there are constraints on the maximum and minimum value of the response. It is of the form:

$$d_i = \begin{cases} \left[\dfrac{Y_i - Y_{i*}}{c_i - Y_{i*}}\right]^s & Y_{i*} \le Y_i \le c_i \\ \left[\dfrac{Y_i - Y_i^*}{c_i - Y_i^*}\right]^t & c_i < Y_i \le Y_i^* \\ 0 & Y_i < Y_{i*} \text{ or } Y_i > Y_i^* \end{cases} \tag{2.7}$$

where:

Y_{i*} is the minimum acceptable value of Y_i,

Y_i^* is the maximum acceptable value of Y_i,

Y_i outside of the range (Y_{i*}, Y_i^*) are unacceptable. c_i is a selected value in the range (Y_{i*}, Y_i^*) in which Y_i is most desirable, and

s, t are shape parameters.

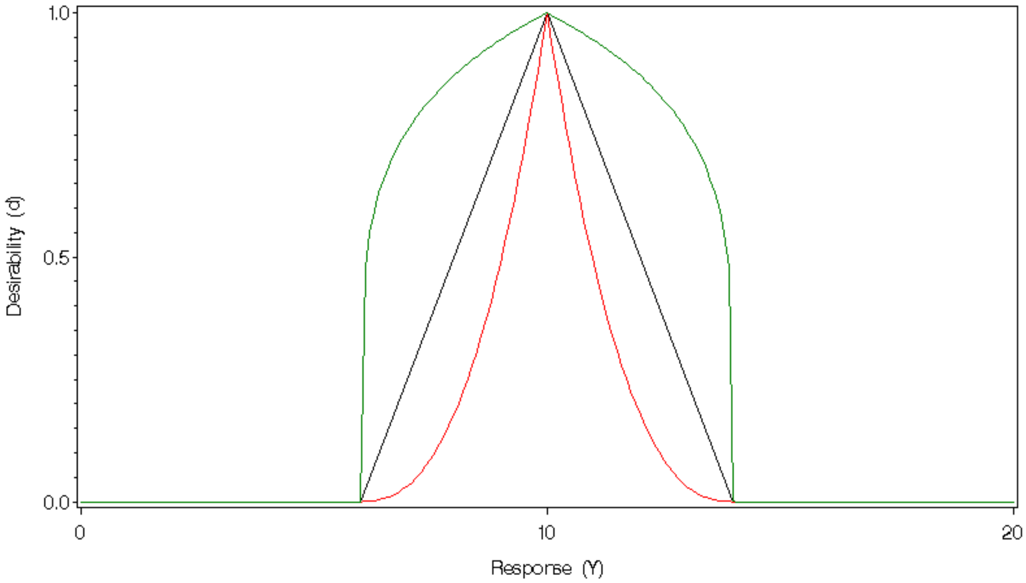


Figure 2.3: Target a response Y .

2.4 Desirability Functions in Other Literature

As previously mentioned, the desirability function approach was created in the quality literature for product manufacturing as a means to balance multiple properties. The desirability function condenses the multivariate problem into a univariate one. In the medical literature, Shih et al. (2003) use the desirability methodology to create a composite measure that is a comprehensive indicator of a patient's outcome status. They use a logistic cumulative distribution function (CDF) for specifying the individual desirability functions. However, any function which maps a response to the (0, 1) interval and which is continuous and differentiable could be used (Shih et al., 2003). Continuous responses were transformed using the logistic cumulative distribution function (CDF) as described in Shih et al. (2003). In the case where the objective is to maximize the response (bigger-is-better), the increasing CDF of the logistic distribution can be expressed as

$$d_i^{\max} = \left[1 + \exp\left(-\left(\frac{Y_i - a_i}{b_i}\right)\right)\right]^{-1}. \tag{2.8}$$

Here a_i is the average of the targeted lower $\left(Y_{i*}\right)$ and upper bound $\left(Y_i^*\right)$ response and b_i controls the spread of the function and is of the form

$$a_i = \left[\frac{Y_i^* + Y_{i*}}{2}\right], b_i = \left[\frac{Y_i^* - Y_{i*}}{2\ln\left(\frac{1-\gamma_i}{\gamma_i}\right)}\right] \text{ where } Y_{i*} < Y_i^* \text{ and } \gamma_i \in (0,1) \text{ such that } d_i\left(Y_{i*}\right) = \gamma_i$$

and $d_i\left(Y_i^*\right) = 1 - \gamma_i$.

For the smaller-is-better case, the decreasing logistic CDF is used and the desirability function can be expressed as

$$d_i^{\min} = \left[1 + \exp\left(\frac{Y_i - a_i}{b_i}\right)\right]^{-1}. \tag{2.9}$$

In this case, γ_i is chosen such that $d_i\left(Y_{i*}\right) = 1 - \gamma_i$ and $d_i\left(Y_i^*\right) = \gamma_i$. The Target Desirability is a combination of the above and is expressed as $d_i^{\text{target}} = d_i^{\max} \times d_i^{\min}$.

Examples of the target (Figure 2.4) and smaller-is-better (Figure 2.5) desirability shapes used by Shih et al. are demonstrated below. In this hypothetical example, the goal is to target a patient's fasting plasma glucose to be within 80-140 mg/dl and for the smaller-is better case we want to minimize the increase in weight that the patient may experience due to the treatment.

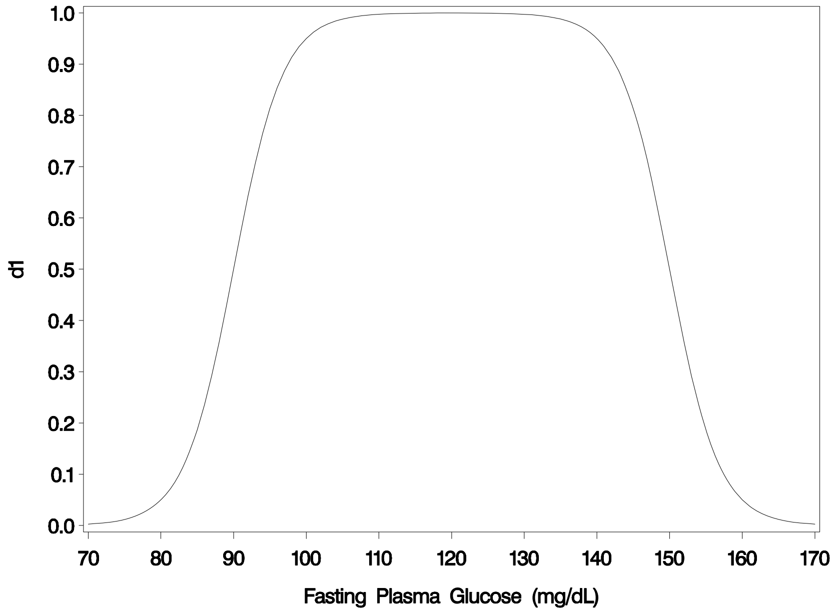


Figure 2.4: Shih et al. (2003) target desirability.

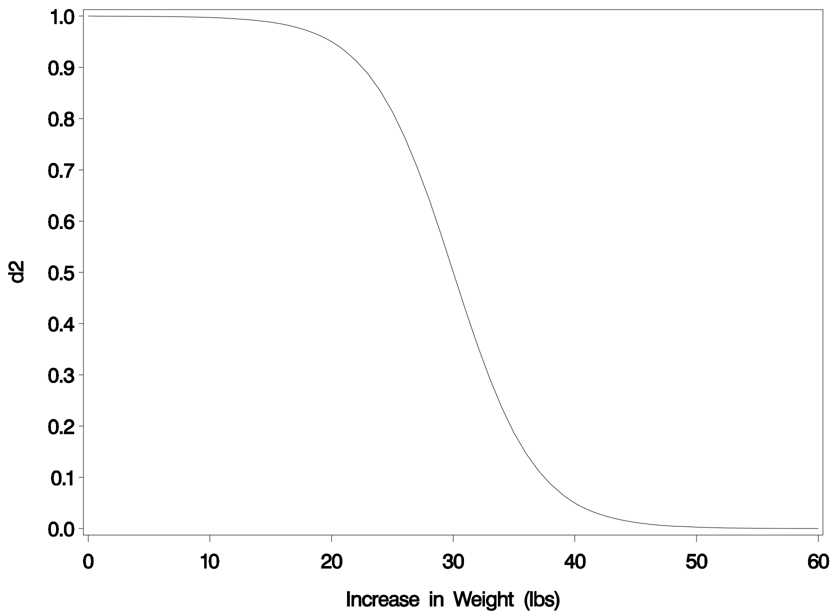


Figure 2.5: Shih et al., (2003) smaller-is-better desirability.

Coffey et al. (2007) demonstrate the use of desirability functions in the toxicology literature where the endpoints are of multiple types, e.g., ordinal, binary, and continuous. A limitation of this approach derives from the fact that the desirability functions and weights must be specified by the user, and thus, there is a degree of subjectivity (Coffey et al., 2007). This is a common problem of other composite scores and that in this case the subjectivity can be minimized by using consensus of expert opinion. Coffey et al. (2007) decrease the subjectivity using questionnaire data. They gathered opinions from a set of neurotoxicologists that had experience in the neurobehaviorial test batteries. A questionnaire was developed that asked the respondents to characterize the level of toxicity indicated by the response levels from each of the endpoints. On a continuum representing no toxicity to most severe toxicity, each respondent was asked to mark a line representing the indicated level of response for each endpoint. An example of the continuum for the endpoint gait score is illustrated in Figure 2.6. The surveys were scored by converting the measured distance between each response and the left boundary of the continuum to a proportion representing the perceived amount of toxicity for each response level (Coffey et al., 2007). There are two main advantages to using desirability function methodology to create a composite score with toxicology data (Coffey et al., 2007). The first advantage is the use of the geometric mean. Using a desirability function that is created with the geometric mean is increasingly sensitive to increasing amounts of toxicity. Secondly, the mechanical incorporation of weights into the geometric mean is simple and intuitive. It allows the endpoints to be prioritized, thus permitting the overall score to give the proper emphasis to each outcome.

Figure 2.6: Sample continuum for gait score.

ENDPOINT: GAIT SCORE

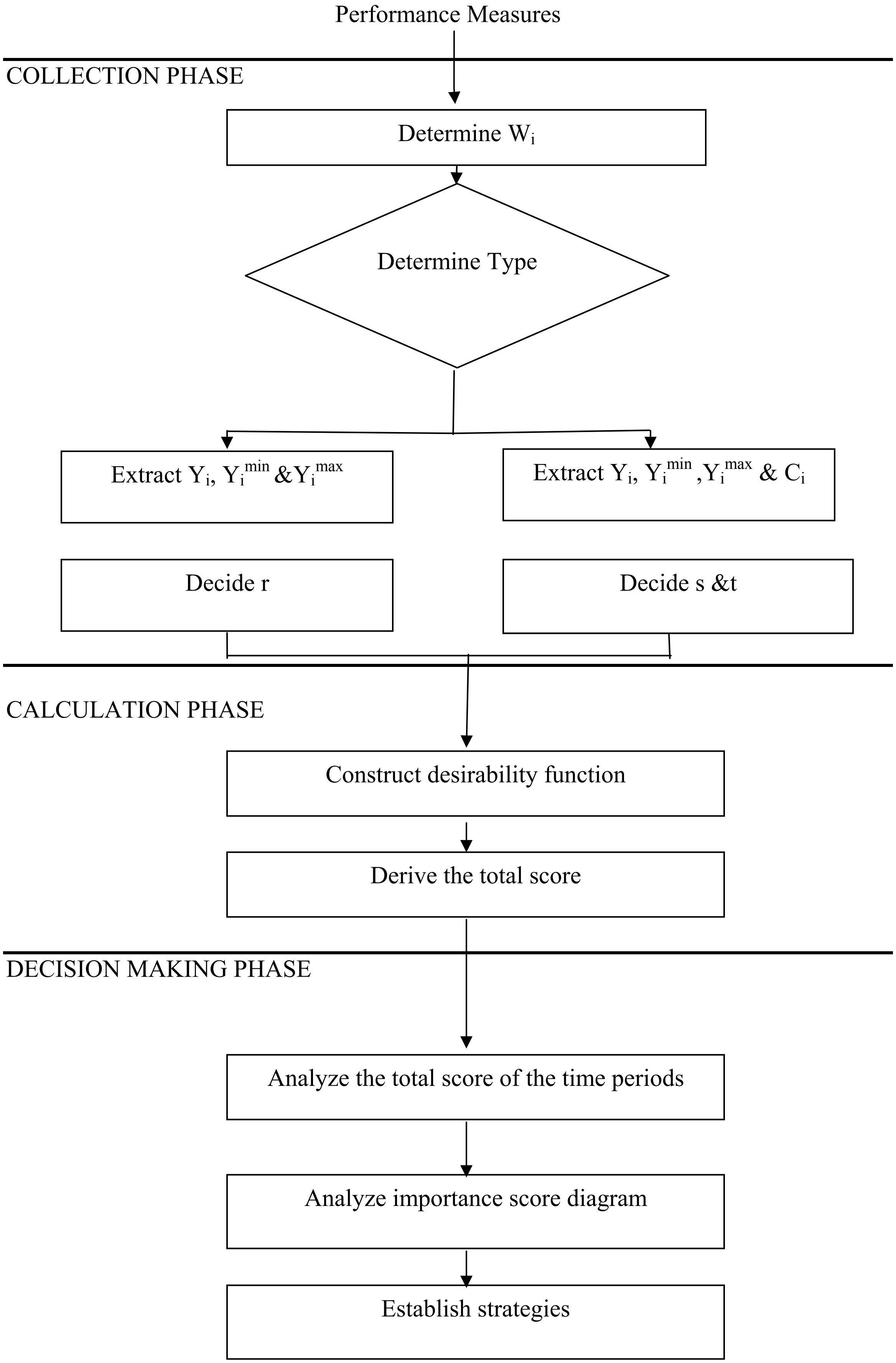
```
                     No                                         Most Serious
                  Toxicity                                        Toxicity
1-none               |--------------------------------------------------|
2-slightly abnormal  |--------------------------------------------------|
3-somewhat abnormal  |--------------------------------------------------|
4-markedly abnormal  |--------------------------------------------------|
5-severly abnormal   |--------------------------------------------------|

                     No                                         Most Serious
                  Toxicity                                        Toxicity
```

Weihs (2007) uses desirability function methodology to combine multiple criteria to identify the best consulting strategy. He develops a quality assurance procedure for the consulting process. The index consists of measurements for quality of results, cost balance, and customer satisfaction.

In the 1990's the balanced scorecard (BSC) was developed as a strategic planning and management system used extensively in business, industry, government, and nonprofit organizations worldwide to align business activities to the vision and strategy of the organization, improve internal and external communications, and monitor organization performance against strategic goals (Kaplan and Norton 2000). It considers both financial and non-financial perspectives and enables companies to track financial results while simultaneously monitoring progress in building the capabilities and acquiring the intangible assets they would need for future growth (Hong and Suh 2005). Hong and Suh (2005) described methodology to calculate a BSC total score based on weighted desirability functions. The total score described the effectiveness of the company. The weights w_i are determined using the Analytic Hierarchy Process (AHP). The individual desirability functions for each of the performance measures are obtained as described in equations (2.5), (2.6) and (2.7). In equation 2.5 the goal is to maximize the response and this corresponds to the larger-the-better type (LTB). Smaller-the-better (STB) corresponds to maximizing the response (eq. 2.6) and targeting a value is the same as the Nominal-the-better in (eq. 2.7). A diagram of the model is shown in Figure (2.7).

Figure 2.7: The Model for Deriving the BSC Total Scorecard


Performance Measures
COLLECTION PHASE
Determine W_i
Determine Type
Extract Y_i, Y_i^{min} &Y_i^{max}
Extract Y_i, Y_i^{min} ,Y_i^{max} & C_i
Decide r
Decide s &t
CALCULATION PHASE
Construct desirability function
Derive the total score
DECISION MAKING PHASE
Analyze the total score of the time periods
Analyze importance score diagram
Establish strategies

In this section we have introduced the concept of desirability functions and its use in different literatures. We have also illustrated different shapes used to transform the responses into individual desirabilities. Each shape must be specified a prior and is best done so by a consensus of experts knowledgeable in the area of interest to reduce subjectivity. Coffey et al. (2007) demonstrated the use of a survey to define the desirability shapes for each of the endpoints. The weighted desirability function was also introduced to enable the assignment of different levels of importance for each of the responses. In addition to the shapes, the weights also need to be specified adding additional subjectivity to the matter. Moreover, it is often difficult to select such weights. In the next chapter we will develop methodology that will find the weight transformations of a composite score such that the score is optimally related to an empirical objective function. We are not aware of other empirical approached for determining optimal shapes of the desirability functions.

CHAPTER 3

Optimal Transformations of a Composite Score

3.1 Introduction

In numerous clinical/experimental studies, multiple endpoints are measured on each subject. It is often not clear which of these endpoints should be designated as of primary importance. This collection of endpoints is often interrelated and measured in various units (Harrington 1965). Coffey et al. (2007) note the inflation of the type one error rate if each endpoint is analyzed independently and suggest combining endpoints into a single unitless composite score. The composite score then becomes an overall assessment of the feature of interest.

Composite scores are common in many aspects today. The National Basketball Association uses composite scores to rank its players. These scores consist of several offensive and defensive components. The Marine Corps uses a composite score to determine promotion. Many attributes are considered such as the marines' conduct mark, rifle score, time in service, educational points, etc. Composite scores have been used in cancer treatment trials to assess a patient's quality of life. The Model for End-Stage Liver Disease (MELD) score is used as a basis for a liver allocation policy. It was developed to assess the short-term prognosis of patients undergoing transjugular intrahepatic portosystemic shunt (TIPS). This score is based on the etiology of the liver disease and three biochemical variables. It is used as a prognostic indicator for patients with advanced chronic liver disease and applied to prioritize patients on the waiting list for a liver transplant. Coffey et al. (2007) introduced such a composite score to the toxicology literature using desirability functions. Harrington (1965), a pioneer in the desirability function literature, states that the critical step in calculating the over-all quality of a product is by establishing the relation between each property and its individual desirability function. He notes that this necessary step is highly subjective as with any other measure of quality/goodness. An advantage of using desirability functions is that multiple data types can be combined. Also since the desirability function is combined using the geometric mean of numbers between 0 and 1, it is sensitive to any one variable being undesirable.

There also exist weighted composite scores where the weights merely establish a new relationship between the property in question and the set of individual desirability functions *d*. However the question remains of how to choose the weights. It is usually difficult to select the weights that measure the relative importance associated with each objective in the weighted sum method (Xu et al., 2004). In this chapter, it is our goal to implement methodology to establish such a relationship empirically rather than subjectively. Iterative algorithms are used to find transformations that optimize an empirical objective criterion.

The notation for the desirability functions are developed in section 3.2. In section 3.3, we introduce a method to determine weights/transformation parameters of a composite score that optimize an empirical objective criterion. In particular, we find the optimal weights/transformations that minimize the generalized variance of a prediction regression model relating the score and response in pre-clinical data; however, other objective criteria can be used.

To implement this method, a pre-clinical example is given. Investigators are conducting studies for development of vaccines for bioterrorism agents. The studies include intensive observational schedules to track the behavior/physiological changes in each animal post exposure to the toxin. It is important to develop a scoring method that can demonstrate worsening conditions with high likelihood of death. With such a tool, investigators can proceed to euthanize animals to decrease pain and suffering resulting in death. Monitoring experimental subjects in vaccine trials with such a tool will provide an objective and focused description of behavior and physiological changes.

3.2 Desirability Function Methodology

In many studies, numerous endpoints are observed simultaneously. The desirability function approach is a method that combines multiple endpoints into an overall score (Harrington 1965). These multiple endpoints can be of different data types (i.e., binary, ordinal, continuous; Coffey et al., 2007). In developing an overall composite score we first develop a score for each of the individual outcome variables. Let y_{ij} $(y_{ij} \in \mathbb{R})$ be the observed value of the j^{th} endpoint $(j = 1, 2, ..., k)$ for the i^{th} subject $(i = 1, 2, ..., n)$. The individual desirability value is then be defined as $d_{ij}(y_{ij}) \in [0,1]$, where d_{ij} is a transformation of the observed score mapping $\mathbb{R} \to [0,1]$.

Once each of k responses has been transformed into its corresponding desirability functions, say $d_{i1}, d_{i2}, ..., d_{ik}$, an overall desirability index can be computed by combining the individual scores through the geometric mean as proposed by Harrington (1965).

$$D_i = \left(d_{i1} \times d_{i2} \times \ldots \times d_{ik}\right)^{1/k} \tag{3.1}$$

Trautmann and Weihs (2006) derive distributions of the index by using the result that the log of the index is additive. The geometric mean is used because it is more sensitive to undesirable outcomes compared to the arithmetic mean (Coffey et al., 2007). That is, since the geometric mean is the product of numbers between 0 and 1, if any of the individual scores is undesirable ($d_{ij} < 1$), then the overall desirability score will be less than one. The geometric mean will be less than or equal to the simple average of the values.

A weighted desirability score is defined where different levels of importance can be assigned to each individual outcome$\left(d_{ij}\right)$. The overall weighted composite score proposed by Derringer (1994) is expressed as:

$$D_i = \left(d_{i1}^{w_1} \times d_{i2}^{w_2} \times \ldots \times d_{ik}^{w_k}\right)^{1/\sum_1^k w_j} . \tag{3.2}$$

In this formulation, the weights w_j, are subjective values given by experts in the particular field of study. In previous use of the desirability functions, such subjectivity is minimized through a consensus of expert opinion (Derringer, 1994). In the following section we address the subjectivity of specifying weights and/or scoring schemes and propose methodology for this task.

3.3 Optimal Transformation Parameters (Method Development)

In this section our goal is to minimize the subjectivity aforementioned by developing methodology to determine the weights or transformation parameters of a composite score that will optimize an empirical objective criterion. Define

$$D_{(opt)i} = \left(f_1 \cdot f_2 \cdots f_k \right)^{1/k} \quad \text{for i} = 1, 2, \ldots n. \tag{3.3}$$

where the transformation function $f_j \in [0,1]$, and can be expressed as a function of the assigned score $f_j\left(d_j\left(y\right)\right)$ or as a direct function of the observed score $f\left(y\right)$. For example, recall the formulation of the desirability index in (3.2) where the individual desirability $\left(d_j\right)$ was assigned weight w_j. Here, the weighted score is viewed as an additional transformation (i.e., Box-Cox transformations) of the assigned score defined as $f_j = f_j\left(d_j\left(y\right)\right) = d_j^{w_j}$ where w_j is the transformation parameter. An example of the direct transformation of the observed score would be to consider a nonlinear logistic function $f_j = f_j\left(y\right) = \left[1 + \exp{-}\left(\beta_{0j} + \beta_{1j} y\right)\right]^{-1}$ where y is the observed score and β_{mj} are transformation parameters for $m = 0$ and 1. Any function mapping the response to the [0,1] scale can be used. Thus, the natural choice is any cumulative distribution function (CDF). The nonlinear logistic function is used for illustration purposes. Also, note that a different transformation f_j can be used for each of the $j = 1,...,k$ endpoints or response variables.

Once desirability values are defined through transformation functions and the empirical model $\mathrm{Z}(\bullet)$ is chosen where we want to study the composite score and some external response variable (ζ), an algorithm is implemented to find optimal transformation parameters. The objective function to be optimized that relates the composite score (D) to the external empirical response (ζ) is defined as $H\left(D\left(f(\beta)\right);\zeta,\theta\right)$. For example, say it is of interest to relate the composite score to time to an event. The Cox regression model can be used to examine this relationship and is written

$$h_i(t) = \exp\{\theta D_i\} h_0(t)$$

where:

D_i is the composite score of behavioral endpoints for the i^{th} subject

θ is the unknown coefficient of the explanatory variable

$h_0(t)$ is the unspecified baseline hazard function at time t.

In this example, $\mathrm{Z}(\bullet)$ is the Cox regression model and (ζ) is time to death or censoring. For illustration, the choice of the objective function $H\left(D\left(f(\beta)\right);\zeta,\theta\right)$ to be optimized could be the generalized variance which is defined as the determinant of the variance-covariance matrix $\left|Var\left(\hat{\theta}\right)\right|$ where $\hat{\theta}$ is the estimate of θ. Moreover, it is our goal to $\min_{\beta} H\left(D\left(f\left(\beta\right)\right);\zeta,\theta\right)$. Given a fixed transformation f_j and initial values for the transformation parameters β, an algorithm may be implemented to optimize the objective criterion.

We propose using a nonlinear optimization subroutine for parameter estimation embedded within a direct search algorithm to find the optimal transformation parameters. The transformation parameters are optimal in the since that we find the transformation parameters that minimizes the variance of the regression parameters. In the iterative process of guiding us to optimal transformation parameters, the values of the composite score change as better parameters are found. The embedded subroutine thus allows iterative calculations of the parameter estimates ($\widehat{\theta}$) as the transformation parameters (β_j) change. This process allows the update of the objective function $H\left(D\left(f\left(\beta\right)\right);\zeta,\theta\right)$, which is to be minimized.

Based on the selected empirical model for analysis $Z\left(\bullet\right)$; the parameter estimates can be found using the usual iterative methods (e.g. Gauss Newton). SAS IML offers a set of optimization subroutines for minimizing/maxmimizing a continuous nonlinear function. For illustration, the subroutine selected was the Newton-Raphson Ridge (nlpnrr) method. For the maximum likelihood estimation the objective function is the log-likelihood function (l) of the empirical model. The use of nlpnrr requires the use of the first and second-order derivatives. If the derivatives are not specified, they can be approximated using the finite difference approximation subroutine (nlpfdd). The user has a choice in using the forward or central difference approximation. In our example we use the forward difference approximation where the first and second order derivatives are approximated by $\frac{\partial l}{\partial x_i} \approx \frac{l\left(x+h_i e_i\right)-f\left(x\right)}{h_i}$ and $\frac{\partial^2 l}{\partial x_i \partial x_j} \approx \frac{l\left(x+h_i e_i+h_j e_j\right)-l\left(x+h_i e_i\right)-l\left(x+h_j e_j\right)-l\left(x\right)}{h_i h_j}$ respectively where h is the step size. Further details can be found in the SAS documentation. Once the parameter estimates are specified and derivatives are approximated the corresponding Hessian matrix can be obtained either from the Hessian module if user specified or directly from the nlpfdd when not specified. The calculations of the variance-covariance matrix then follows $V=\left(I\right)^{-1}$ where the Information matrix $\left(I\right)$ is the negative of the Hessian matrix. The generalized variance is defined as $\left|\hat{\mathbf{V}}\left(\hat{\theta}\right)\right|$.

The Nelder-Mead algorithm is used to find the transformation parameters that minimize the generalized variance. The Nelder-Mead algorithm is a method that is based on evaluating a function at the vertices of a simplex, then iteratively shrinking the simplex as better points are found until some desired bound is obtained (Nelder and Mead 1965). When there are n variables being optimized, the simplex consists of n+1 vertices. For example, for two variables, the simplex is a triangle and the search method compares the function value at the three points. Iteratively, the triangle moves away from the worst point where the function value is the largest and generates triangles in which the function value at the vertices get smaller until the optimum points are found.

A procedural outline of the methodology is as follows:

Preliminary Steps

1. Assign scores for individual desirability function.
    a. If endpoints are ordinal or binary assign scores accordingly.
    b. If endpoints are continuous, determine if the objective is to maximize/minimize or reach a target value and assign initial shapes accordingly.
    c. Choose the type of transformation function $f_j\left(d_j\left(y\right)\right)$ for the desirability calculation.
2. Select external variable $\left(\zeta\right)$ and empirical model $Z\left(\bullet\right)$ (e.g., Cox regression, logistic regression) relating $\left(\zeta\right)$ and composite score D.
3. Select the objective function $H\left(D\left(f\left(\beta\right)\right);\zeta,\theta\right)$ to be optimized.

Initialization Step

1. Find initial parameter estimates β_{0j}, β_{1j} for calculations of transformation function $f_j = f_j\left(d_j\left(y\right)\right)$ such that $D^0_{(opt)i} = \left(f_1 \cdot f_2 \cdots f_k\right)^{1/k}$.
2. Initialize starting value $\boldsymbol{\theta}$ for maximum likelihood estimation in the optimization subroutine.
    a. Specify the form of the objective function (i.e. log-likelihood) for the maximum likelihood estimation.

b. Specify starting values for the maximum likelihood estimates.

c. Determine if user will use the finite difference method for derivative approximations or write a Hessian module specifying its form.

Algorithm

Step 1: Evaluate the objective function $H\left(D^s\left(f\left(\beta\right)\right);\zeta,\theta\right)$.

Step 2: Use a direct search algorithm (e.g. Nelder-Mead) to find β_j^{s+1} and corresponding D_i^{s+1}.

Step 3: Find $\hat{\theta}^{s+1}$ using an optimization subroutine, e.g. nlpnrr in SAS/IML.

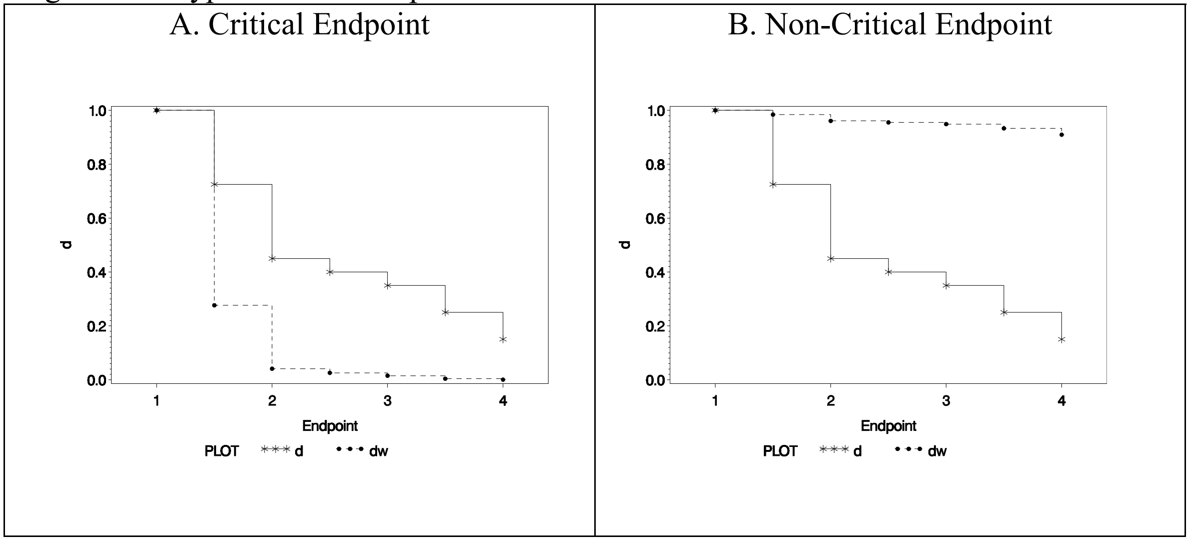
Step 4: Repeat steps 1-3 until convergence, i.e. $\min_f H\left(D\left(f\left(\beta\right)\right);\zeta,\theta\right)$.

Step 5: Repeat steps 1-4 with multiple step sizes and starting values and select best case (i.e. case with minimum variance).

3.4 Penalized Optimality Methodology

When searching for statistical optimality, the values of the resulting characterizations of the optimal transformation parameters may not be in agreement with expert judgment. Optimized parameters are found such that a given statistical objective criterion is satisfied; thus, values found may lead to impractical consequences. For example, consider the Box-Cox transformation $D_{(opt)} = \left(f_1 \cdot f_2 \cdots f_k\right)^{1/k}$ where $f_j = f_j\left(d_j\left(y\right)\right) = d_j^{w_j}$ for $j = 1, 2, \ldots k$. The transformation parameter of interest is w_j. The optimal value for w_j will result in steepening the transformation curve for critical endpoints (Figure 3.1 A). For less critical endpoints, the value of w_j lessens the relative importance and transformation curves are more nearly horizontal (Figure 3.1 B). Consider the case where there are five endpoints (k=5) combined in the score. Suppose optimal values of the transformation parameters (w_j) are found and relative importance is only placed on one of the five endpoints.

Figure 3.1: Hypothetical example of transformation effects



Researchers with expert judgment may not agree and say from prior experience that some importance should be placed on other endpoints. In this section we propose methodology that will combine empirical optimality and expert judgment using a similar strategy to the penalized optimization criterion described by Parker and Gennings (2008). The desirability functions are used to penalize impractical transformations. Define a desirability function for the penalty as $D_{(Penalty)i} = \left(d_{i1} \times d_{i2} \times .....d_{iq}\right)^{1/q}$ as in 3.1 for the q characteristics under consideration. The penalty function is then represented by$\left(1 - D_{(Penalty)i}\right)$. Let $H\left(D\left(f(\beta)\right);\zeta,\theta\right)$ be the value of the optimality criterion i.e., generalized variance as described in Section 3.3. The penalty function may be added to the optimality criterion to penalize indices that may otherwise be deemed empirically optimal, yet unacceptable based on expert opinion. The penalty function takes on values between 0 and 1 where a value of 1 indicates poor agreement with expert opinion. A user defined scaling constant Λ is used to control the weight of the penalty function relative to the optimality criterion. Thus for a given Λ, a penalized optimal index which jointly minimizes $H\left(D\left(f(\beta)\right);\zeta,\theta\right)$ and $\left(1 - D_{(Penalty)i}\right)$ is defined as:

$$H\left(D\left(f(\beta)\right);\zeta,\theta\right) + \Lambda\left(1 - D_{(Penalty)i}\right). \tag{3.4}$$

In choosing a value of Λ, Parker et al. suggest to initially set $\Lambda = \min H\left(D\left(f(\beta)\right);\zeta,\theta\right)$, the minimum value of the optimality criterion and minimize eq. (3.4). Multiples of the minimum values are then considered for Λ (i.e. $\Lambda = l \min H\left(D\left(f(\beta)\right);\zeta,\theta\right)$ where l is a positive number. Parker et al. recommend choosing the final value of Λ in the range where there is stability in the desirability function and the optimality criterion

3.5 Application of methodology

3.5.1 Background

To illustrate the development of this methodology we use data from a Botulinum study. The Botulinum neurotoxins are considered to be the deadliest naturally occurring toxins known to man. It is listed by the Centers for Disease Control and Prevention as a category A bioterrorism agent where these agents pose highest risk to the public and national security, thus studies are done to produce animal models to better understand such agents. Also, guidelines from the NIH (1996) for laboratory animal care dictate the use of data-based criteria that are predictive of impending death to implement timely euthanasia cases where the pain and distress category is E. The morbidity evaluation of these animals involves multivariate data including observational, biological and behavioral variables measured repeatedly. In this example we develop a morbidity composite score using methodology based on desirability functions similar to Coffey et al. (2007) with validation of the composite score based on its statistical relationship to instantaneous hazard of death. Optimal transformation parameters are found and the methods are compared.

3.5.2 Data Summary

In a study conducted by scientist at the Lovelace Respiratory Research Institute, endpoints were examined on female CD-1 mice exposed to Botulinum toxin B in 5 dose groups with 10 mice per group which were monitored up to twice daily for 5 days. On day one scores were only taken in the afternoon and on day five only in the morning. Ordinal scores for piloerection (present=1/not present=2), muscle tone (normal=1, moderate loss=2, severe loss=3), respiration (normal=1, thoracic tachypnea=2, abdominal tachypnea=3, dyspnea=4), and activity (normal=1, decreased=2, little or no activity=3) were taken on each animal.

Table 3.1: Summary of Response

Dose ng/kg	Number of Deaths	Total Exposed
0.7	0	10
1	0	10
3	0	10
5	5	10
7	9	10

A summary of the animals' response (death) by dose groups is given in Table 3.1. Deaths were only experienced at the two highest doses (5 ng/kg , 7ng/kg). There were a total of 14 deaths and the remaining 36 animals were euthanized at study end.

The frequencies of the observed scores for all endpoints over time are displayed in Table 3.2. For each endpoint, the majority (at least 65%) of the scores are normal (observed score $x_{ij} = 1$). When focusing on the animals' last observation prior to death or censoring for each endpoint (see Table 3.2) the majority of the scores (at least 56%) are still normal. However, for those that died (14/50) only 29% at most have normal scores.

To study the endpoints that may be indicative of toxicity, we closely examine and summarize the values of the observed scores for the animals that died (Table 3.4). Out of the nine deaths for the highest dose group (7 ng/kg), three animals die too rapidly. That is, they die or are euthanized prior to the second morning when scores are recorded again thus only having scores from the first afternoon prior to showing symptoms of toxicity with normal scores. For activity, 4/14 animals that died displayed the most severe outcome (observed score of 3) and 8/14 had a score of at least 2.5. In the case of the endpoints respiration and muscle tone, there were no animals to display the most severe outcome. For respiration, 8/14 had scores of at least 3.0 where 3/14 were 3.5. Only 3/14 animals that died had a score of 2.5 for muscle tone. Of the 14 animals that died, nine had piloerection at their time of death and two additional animals had scores of 1.5. However, there were eleven animals that did not die that also have scores of at least 1.5 for piloerection.

Table 3.2: Total Frequency of Scores and Frequency of Scores at Subjects Last Observation Prior to Death

Endpoint	Observed Score	Total Frequency	Total Percent	Frequency Censored	Frequency Non-Censored
Activity					
	1	307	94.75	36	4
	1.5	7	2.16	0	2
	2	2	0.62	0	0
	2.5	4	1.23	0	4
	3	4	1.23	0	4
Respiration					
	1	215	66.36	32	2
	1.5	39	12.04	3	0
	2	44	13.58	1	1
	2.5	8	2.47	0	2
	3	15	4.63	0	6
	3.5	3	0.93	0	3
	4.0	0	0	0	0
Muscle Tone					
	1	279	86.11	33	3
	1.5	23	7.10	2	0
	2	19	5.86	1	8
	2.5	3	0.93	0	3
	3.0	0	0	0	0
Piloerection					
	1	255	78.70	25	3
	1.5	37	11.42	9	2
	2	32	9.88	2	9

Table 3.3: Last Scores for Animals That Died

Animal ID	Dose ng/kg	Activity[a]	Respiration[b]	Muscle Tone[c]	Piloerection[d]
M021	5	3	3.5	2.5	2
M022	5	2.5	3.0	2.0	2
M025	5	1.5	3	2	2
M026	5	1	3	2	2
M030	5	1.5	2.5	2	2
N021	7	1	1	1	1
N022	7	1	2	1	1
N023	7	3	3.5	2.5	2
N025	7	1	1	1	1
N026	7	3	3	2	1.5
N027	7	3	3.5	2.5	2
N028	7	2.5	3	2	2
N029	7	2.5	3	2	2
N030	7	2.5	2.5	2	1.5

[a] activity (normal=1, decreased=2, little or no activity=3)
[b] respiration (normal=1, thoracic tachypnea=2, abdominal tachypnea=3, dyspnea=4)
[c] muscle tone (normal=1, moderate loss=2, severe loss=3)
[d] piloerection (present=1/not present=2)

3.5.3 Creation of Morbidity Score and Statistical Analysis

The initial phase of creating the morbidity score is to define the individual desirability functions ($d_j \in [0,1]$) for each response. Given the ordinal and categorical nature of these variables, the values for each endpoint, activity (d_1), respiration (d_2), muscle tone (d_3), piloerection (d_4) are step functions that were initially defined based on collaboration with scientists at the Lovelace Respiratory Research Institute conducting the study as follows (See Figure 3.2-3.5):

$$d_1 = \begin{cases} 1 & \text{if } x = 1 \text{ (Normal)} \\ 0.8 & \text{if } x = 2 \text{ (Decreased)} \\ 0.5 & \text{if } x = 3 \text{ (Little or None)} \end{cases}$$

$$d_2 = \begin{cases} 1 & \text{if } x = 1 \text{ (Normal)} \\ 0.45 & \text{if } x = 2 \text{ (Thoracic Tachypnea)} \\ 0.35 & \text{if } x = 3 \text{ (Abdominal Tachypnea)} \\ 0.15 & \text{if } x = 4 \text{ (Dyspnea)} \end{cases}$$

$$d_3 = \begin{cases} 1 & \text{if } x = 1 \text{ (Normal)} \\ 0.8 & \text{if } x = 2 \text{ (Mild Loss)} \\ 0.75 & \text{if } x = 3 \text{ (Moderate Loss)} \\ 0.4 & \text{if } x = 4 \text{ (Severe Loss)} \end{cases}$$

$$d_4 = \begin{cases} 1 & \text{if } x = 1 \text{ (Not Present)} \\ 0.8 & \text{if } x = 2 \text{ (Present)} \end{cases}$$

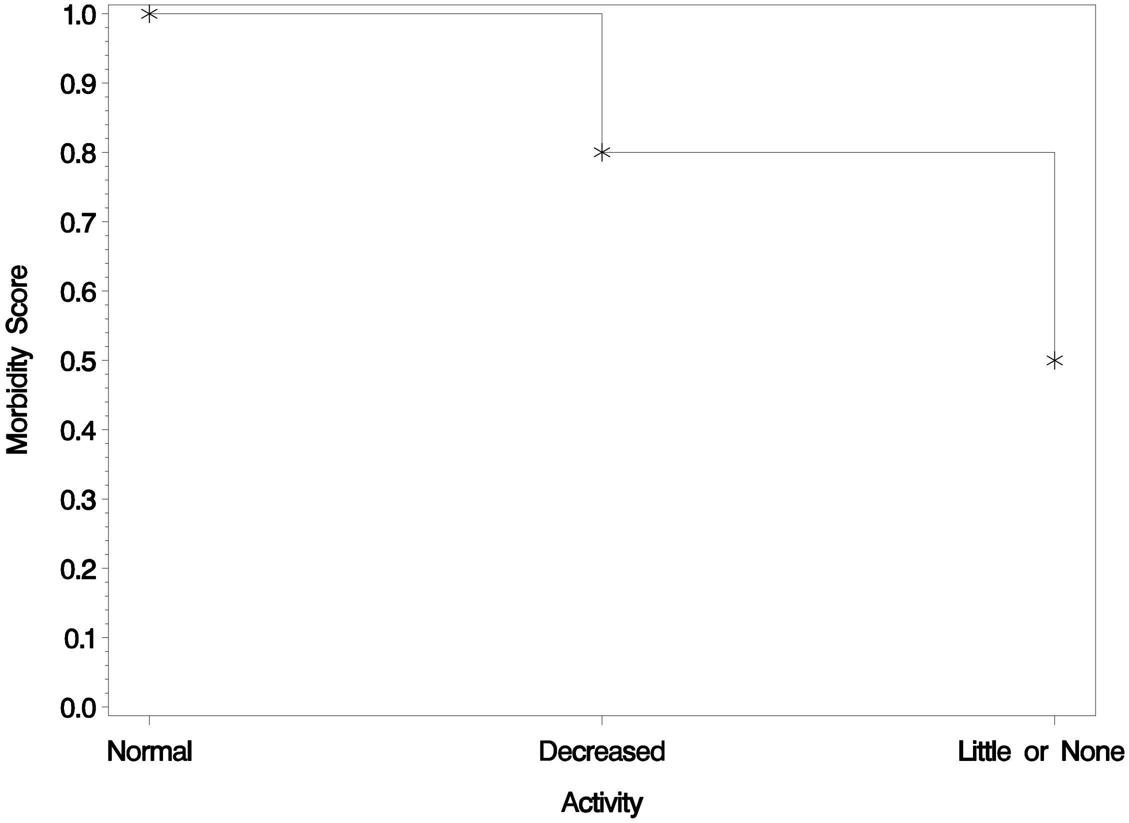


Figure 3.2 Initial Desirability function for activity response

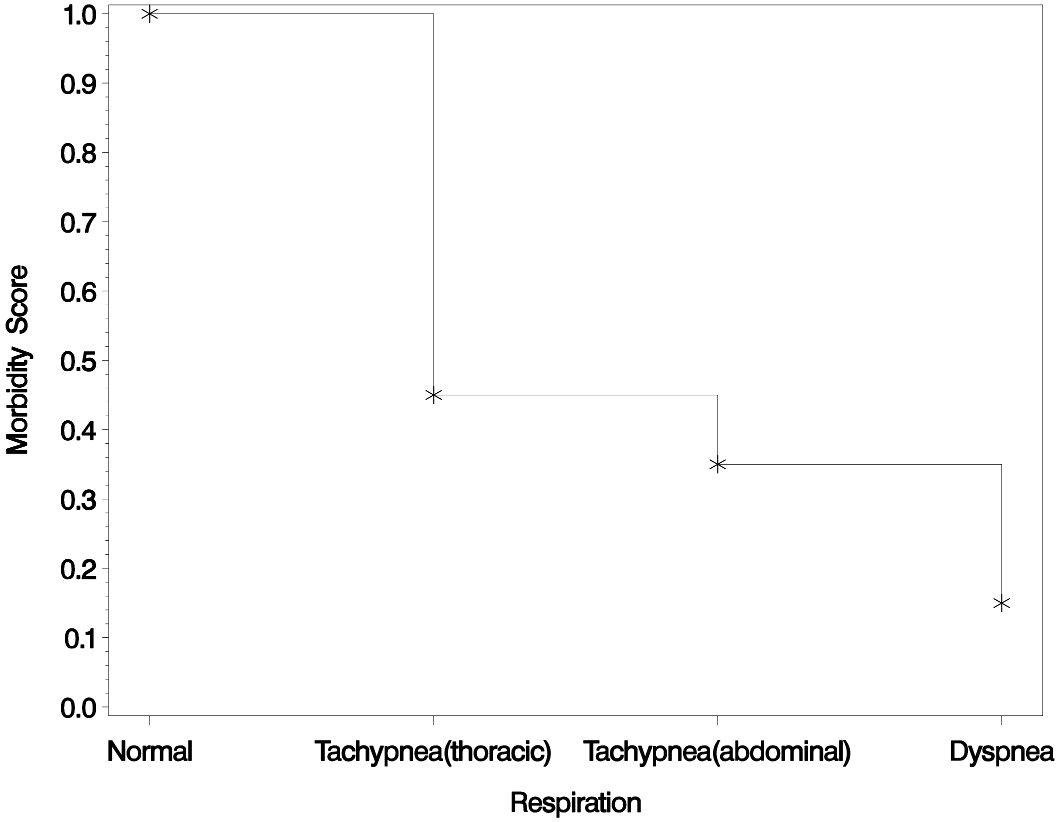


Figure 3.3 Initial Desirability function for respiration response

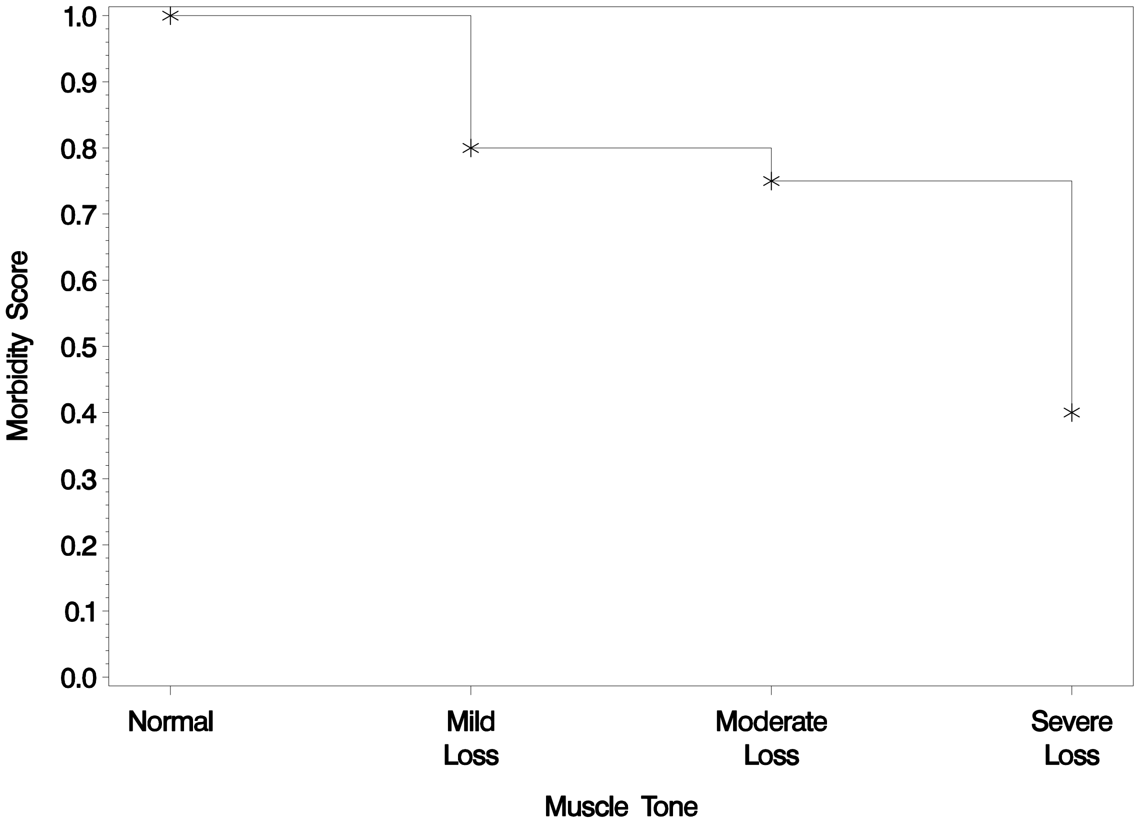


Figure 3.4 Initial Desirability function for muscle tone response

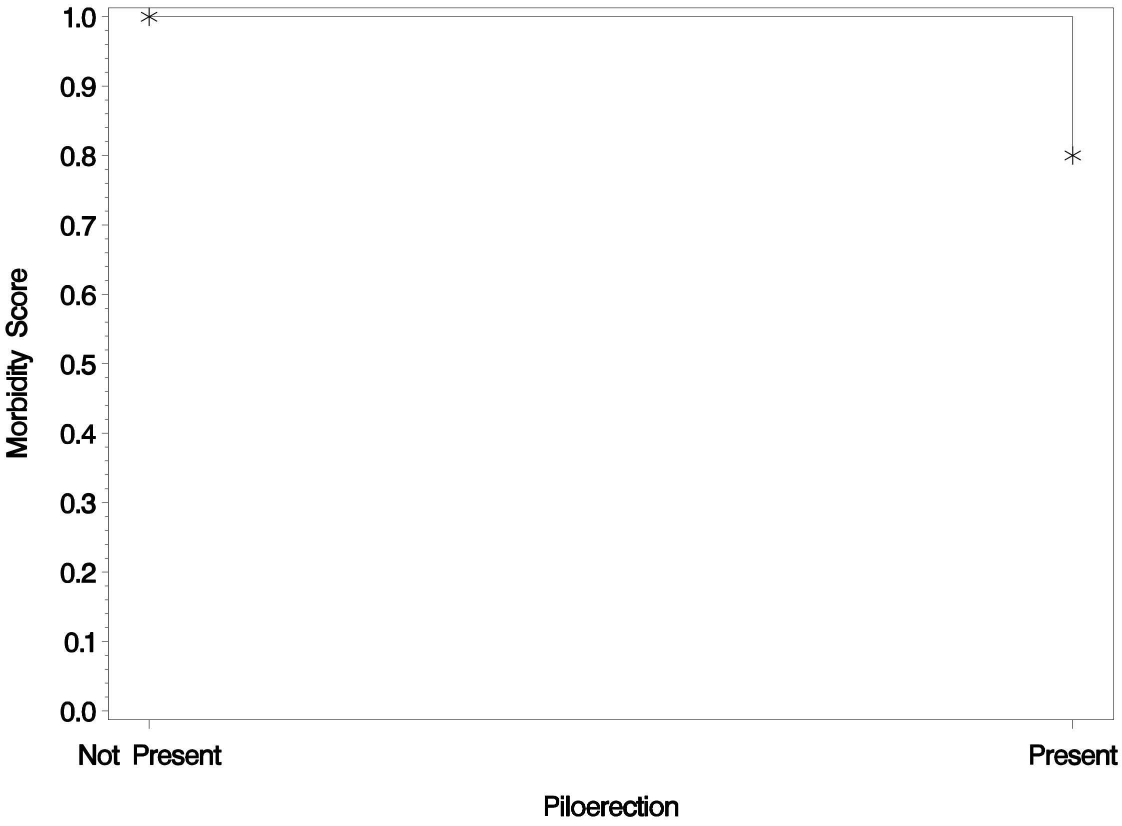


Figure 3.5 initial Desirability function for piloerection response

Higher scores of activity, respiration, muscle tone and piloerection were assigned lower desirability scores. The individual desirability scores were aggregated through the geometric mean shown in (3.1) and a morbidity score for each individual animal was created. One of the objectives of this study was to determine the relation between the morbidity score and hazard of death, thus a Cox's proportional hazards model was used. The Cox's model is widely used in failure-time data to study the relationship of hazard of death with explanatory variables. The Cox regression model with time dependent covariates can be expressed as (Collett 2003):

$$h_i\left(t\right) = \exp\left\{\sum_{l=1}^{p} \theta_l D_{li}\left(t\right)\right\} h_0\left(t\right) \tag{3.5}$$

In this model, $D_{li}\left(t\right)$ is the l^{th} explanatory variable for the i^{th} subject at time t. The unknown coefficients of the explanatory variables are defined as θ_l and $h_0\left(t\right)$ is the unspecified baseline hazard function when the explanatory variables are zero. In this case, our explanatory variable ($D_{li}\left(t\right)$ is doubly bounded between 0, 1 and never reaches zero thus the comparison becomes an extrapolation. For this reason we will model $D^* = 1 - D$. Statistical inference on θ requires at each uncensored time point T_i the values of the covariates for all subjects at risk at time T_i. Given the nature of time-dependent data, depending on the time in which the observation was recorded, the information may not be available. In this situation, if there is a value for this variable prior to and after the time point of interest one could linearly interpolate a value or choose the value closest to that time. When there are only values for that variable prior to the time required one may use the last recorded value for that variable for an individual. This method is called “last value carried forward” (LVCF) and is used in this analysis.

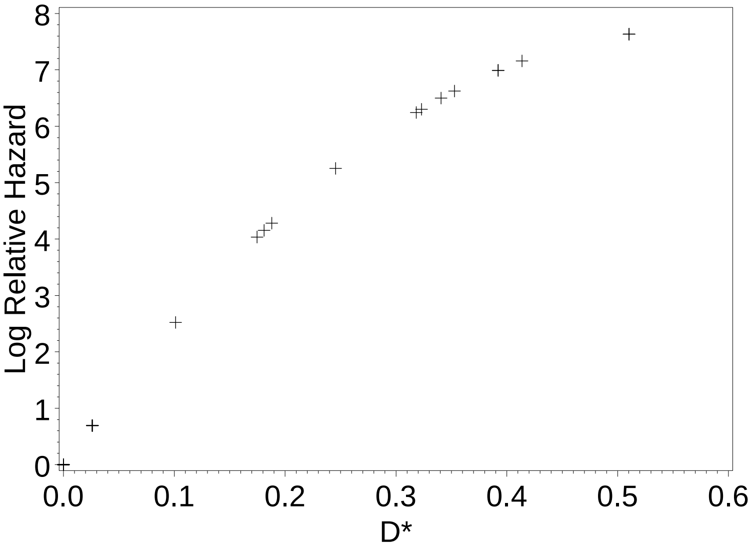
The Cox’s regression model with time dependent covariates was fit to the data using SAS version 9.1.3. The parameter estimates are shown in Table 3.4.

Table 3.4 Parameter estimates for the Cox Regression model (3.4)

Parameter	**Estimate**	**SE**	**Chi-Sq**	**P-value**
D*	27.35	7.93	11.8 9	0.0006
$D*^2$	-24.29	11.76	4.26	0.039

A significant relationship (Likelihood ratio χ^2=44.4, DF=2, p<0.001) was found between the scaled morbidity score (D*) and hazard of death. The morbidity score was significant in a nonlinear (quadratic) manner. As the scaled morbidity score (D*) worsens, there is an increase in log relative hazard which diminishes as D^* increases (Figure 3.6).

Figure 3.6: Log relative hazard vs. Scaled morbidity score



3.5.4 Optimal Estimation

Recall in section 3.3 where the transformation functions were described as a function of the assigned scores, $f_j = f_j\left(d_j\left(y\right)\right)$, or as a direct function of the observed scores, $f_j = f_j\left(y\right)$. The aggregate of the morbidity score is defined as $D_{(opt)i} = \left(f_1 \cdot f_2 \cdots f_k\right)^{1/k}$ for the $k = 4$ endpoints. Here, we use the optimization methodology to find the optimal transformation parameters that minimizes the generalized variance of the Cox-regression model. To implement the algorithm described in section 3.3 we need to specify

1) the form of the likelihood or log-likelihood for the maximum likelihood estimation of the Cox Regression model and

2) the form of the variance-covariance matrix to obtain the generalized variance.

The Breslow approximation to the likelihood function of the Cox regression model is expressed as (Collett, 2003):

$$L(\beta)=\prod_{j=1}^{r}\frac{\exp\left(\boldsymbol{\theta}'\mathbf{s}_j\right)}{\left\{\sum_{l\in R\left(t_{(j)}\right)}\exp\left(\boldsymbol{\theta}'\mathbf{D}_l\right)\right\}^{\delta_j}} \tag{3.6}$$

for r failure times. To incorporate tied survival times, $\mathbf{s}_j$ is defined as the vector of sums of each of the p covariates for those individuals who die at the j^{th} death time $t_{(j)}$ and δ_j is the number of deaths at time $t_{(j)}$ for $j=1,...,r$. The unknown θ parameters are estimated using maximum likelihood methodology. The maximum likelihood estimates (MLE's) of $\boldsymbol{\theta}$ are those estimates $\hat{\boldsymbol{\theta}}$ that maximize the likelihood function ($L(\boldsymbol{\theta})$) and equivalently the log likelihood function $\left(\log L(\boldsymbol{\theta})\right)$. The likelihood function can be expressed as:

$$\log L(\theta)=\sum_{j=1}^{r}\left[\theta' s_j-\delta_j\log\sum_{l\in R\left(t_{(j)}\right)}\exp\left(\theta' D_l\right)\right]. \tag{3.7}$$

The MLE's for the p parameters $\theta_1,\theta_2,.....\theta_p$ are the simultaneous solution to setting the score function to zero:

$$\frac{\partial \log L(\boldsymbol{\theta})}{\partial \theta_1} = 0$$

$$\frac{\partial \log L(\boldsymbol{\theta})}{\partial \theta_2} = 0$$

$$\begin{array}{l} . \\ . \\ . \end{array} \qquad (3.8)$$

$$\frac{\partial \log L(\boldsymbol{\theta})}{\partial \theta_p} = 0$$

where the derivative is of the form

$$\frac{\partial \log L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{j=1}^{r} \left[s_j - \delta_j \left[\frac{\sum_{l \in R\left(t_{(j)}\right)} \exp\left(\boldsymbol{\theta}' D_l\right) \cdot D_l}{\sum_{l \in R\left(t_{(j)}\right)} \exp\left(\boldsymbol{\theta}' D_l\right)} \right] \right] \qquad (3.9)$$

Conditional on D, the variance-covariance matrix for the p parameters are estimated by using the Hessian matrix of second derivatives. Define the observed information matrix $I\left(\hat{\theta}\right)_{p \times p}$ as a matrix of negative second derivatives of the log likelihood function:

$$I(\boldsymbol{\theta}) = \begin{pmatrix} -\frac{\partial^2 \log L(\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_1} & \cdots & -\frac{\partial^2 \log L(\boldsymbol{\theta})}{\partial \theta_p \partial \theta_1} \\ \vdots & \ddots & \vdots \\ -\frac{\partial^2 \log L(\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_p} & \cdots & -\frac{\partial^2 \log L(\boldsymbol{\theta})}{\partial \theta_p \partial \theta_p} \end{pmatrix}_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}} \qquad (3.10)$$

and has the form

$$I\left(\boldsymbol{\theta}\right)=\sum_{j=1}^{r}\left[-\delta_{j}\left[\frac{\sum\limits_{l\in R\left(t_{(j)}\right)}\exp\left(\boldsymbol{\theta}'\mathbf{D}_{l}\right)\cdot\mathbf{D}_{l}\cdot\mathbf{D}_{l}'}{\sum\limits_{l\in R\left(t_{(j)}\right)}\exp\left(\boldsymbol{\theta}'\mathbf{D}_{l}\right)}-\left(\frac{\sum\limits_{l\in R\left(t_{(j)}\right)}\exp\left(\boldsymbol{\theta}'\mathbf{D}_{l}\right)\cdot\mathbf{D}_{l}}{\sum\limits_{l\in R\left(t_{(j)}\right)}\exp\left(\boldsymbol{\theta}'\mathbf{D}_{l}\right)}\right)\left(\frac{\sum\limits_{l\in R\left(t_{(j)}\right)}\exp\left(\boldsymbol{\theta}'\mathbf{D}_{l}\right)\cdot\mathbf{D}_{l}}{\sum\limits_{l\in R\left(t_{(j)}\right)}\exp\left(\boldsymbol{\theta}'\mathbf{D}_{l}\right)}\right)'\right]\right]_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}.\quad(3.11)$$

It follows that the estimated variance-covariance matrix of $\hat{\boldsymbol{\theta}}$ is given by

$\hat{\mathbf{V}}\left(\hat{\boldsymbol{\theta}}\right)=\left[I\left(\hat{\boldsymbol{\theta}}\right)\right]^{-1}$. The generalized variance then is defined as the determinant of the

variance-covariance matrix $\left|\hat{\mathbf{V}}\left(\hat{\boldsymbol{\theta}}\right)\right|$.

3.5.5 Optimal Transformations

Listed in Table 3.5 are the functions considered for this example to transform the observed score onto the 0-1 scale. That is, these functions are used to create partial desirabilites for the i^{th} subject and the j^{th} outcome variable. The first transformation considered is the Box-Cox represented as $f_j = f_j\left(d_j\left(x\right)\right)=d_{ij}^{\beta_j}$. In this function, the weight $\beta_j > 0$ is viewed as an additional transformation of the individual desirability score (d_{ij}). Two nonlinear transformation functions presented are, specifically the logistic and Gompertz functions. In each case, the corresponding CDF was used to transform the observed score to the 0-1 scale. Transformation parameters β_{0j} and β_{1j}, determine the severity of the response. The last transformation type considered is described as the optimum scale transformation. In this case, we directly find the optimum individual desirability value. The normal responses are set at a value of one

($\beta_{jm} = 1$) and all other values are determined from the algorithm. For the logistic and Gompertz transformations, a different objective criterion had to be used due to the scaling of the morbidity score. In this case we chose to maximize the sum of the Wald statistic on each model parameter: $\left(\frac{\theta_1}{SE(\theta_1)}\right)^2 + \left(\frac{\theta_2}{SE(\theta_2)}\right)^2$. The values of the optimal transformation parameters are given in Table 3.8.

Table 3.5: Examples of Transformation Functions

Box-Cox	$f_j = f_j\left(d_j\left(x\right)\right) = d_{ij}^{\beta_j}$
Logistic	$f_j(x) = \frac{1}{1+\exp\left(-\beta_{0j} + \beta_{1j}x_{ij}\right)}$
Gompertz	$f_j(x) = \exp\left(-\exp\left(-\beta_{0j} + \beta_{1j}x_{ij}\right)\right)$
Optimum Scale	$f_j(x) = \begin{cases} 1 & \text{if } x_j = 1 \\ \beta_{jm} & \text{otherwise } j = 1,....,4 \\ & \qquad m = 1,....,C_j - 1 \end{cases}$

Table 3.6: Parameter Estimates

Model	Parameter	Estimate	SE	Chi-Sq	P-value	Criterion	LR 2DF Chi-sq	P-value
1	D*	27.35	7.932	11.89	0.0006	1028.28* 16.15**	44.43	<.0001
	D^{*2}	-24.29	11.763	4.264	0.0389			
2	D*	2.278	4.449	0.262	0.609	81.718*	39.05	<.0001
	D^{*2}	7.798	5.637	1.914	0.116			
3	D*	38.29	9.27	17.06	<0.001	30.64**	46.80	<.0001
	D^{*2}	-31.39	8.52	13.58	<0.001			
4	D*	22.502	5.138	19.18	<0.001	34.078**	37.80	<.0001
	D^{*2}	-21.477	5.561	14.91	<0.001			
5	D*	8.667	2.644	10.75	0.001	5.199*	33.88	<.0001
	D^{*2}	-4.103	2.410	2.90	0.089			

Note: Model 1: Assigned Scores, Model 2: Box-Cox, Model 3: Logistic,
Model 4: Gompertz, Model 5: Optimal Scale
*Generalized Variance, **Sum of Wald Statistic

Table 3.7: Optimal Transformation Parameters

	Box-Cox	**Logistic**	**Gompertz**	**Scale**
Activity	$\beta_1 = 0.00019$	$\beta_{01} = 9.99$ $\beta_{11} = -6.45$	$\beta_{01} = 9.032$ $\beta_{11} = -2.116$	$\beta_{11} = 0.9814$ $\beta_{12} = 0$
Respiration	$\beta_2 = 3.998$	$\beta_{02} = 1.82$ $\beta_{12} = -1.26$	$\beta_{02} = 9.998$ $\beta_{12} = -3.324$	$\beta_{21} = 0.9473$ $\beta_{22} = 0.8726$ $\beta_{23} = 0.0262$
Muscle Tone	$\beta_3 = 0.00055$	$\beta_{03} = 4.75$ $\beta_{13} = -0.80$	$\beta_{03} = 4.527$ $\beta_{13} = -0.106$	$\beta_{31} = 0.7013$ $\beta_{32} = 0.0099$
Piloerection	$\beta_4 = 0.00102$	$\beta_{04} = 4.74$ $\beta_{14} = -0.023$	$\beta_{04} = 7.6498$ $\beta_{14} = -3.458$	$\beta_{41} = 0.0497$

As expected, in the nonlinear functions the slope parameters (Table 3.7) were all negative. But the dominating parameter was the intercept where large values (say >3 or 4) are associated with shapes with long plateaus. The small Box-Cox parameters are associated with less important endpoints. Given the optimal transformation parameters for the morbidity index (Table 3.7), the corresponding Cox regression models were analyzed (Table 3.6). In all cases, there was a significant relationship ($p<0.001$) between the morbidity score and hazard of death.

In comparison to the assigned scores, the Box-Cox and optimal scale transformations reduced the generalized variance from 1028.28 to 81.2 and 5.2 respectively. Because different criterion was used, direct comparisons of the optimization criterion between the Box-Cox and the nonlinear transformation functions could not be made. The effects of the transformation parameters given in Table 3.7 can be seen in Table 3.8. The Box-Cox method put sole importance on endpoint respiration. The Gompertz transformation indicated relative importance for endpoints respiration and piloerection. The optimum scale method was the only method that signaled importance for each of the endpoints.

Table 3.8: Transformed Values

	Assigned Scores	**Box-Cox**	**Logistic**	**Gompertz**	**Opt. Scale**
Number of Parameters		4	8	8	8
d1 Motor Activity					
Normal	1	1	0.97	1	1
	0.9	1	0.58	1	0.99
Decreased	0.8	1	0.05	1	**0.98**
	0.65	1	0.002	0.98	0.49
Little or None	0.5	1	0	0.93	**0**
d2 Respiration					
Normal	1	1	0.64	1	1
	0.725	0.28	0.48	0.99	0.97
Tachypnea (thoracic)	0.45	0.04	0.33	0.97	**0.95**
	0.4	0.03	0.21	0.83	0.91
Tachypnea (abdominal)	0.35	0.02	0.12	0.38	**0.87**
	0.25	0	0.07	0.01	0.45
Dyspnea	0.15	0	0.04	0	**0.03**
d3 Muscle Tone					
Normal	1	1	0.98	0.99	1
	0.9	1	0.97	0.99	0.85
Moderate Loss	0.75	1	0.96	0.99	**0.7**
	0.575	1	0.94	0.99	0.36
Severe Loss	0.4	1	0.91	0.91	**0.01**
d4 Piloerection					
Not Present	1	1	1	0.98	1
	0.9	1	1	0.92	0.52
Present	0.8	1	1	0.62	**0.05**

In all cases there was a significant relationship between the morbidity score and hazard of death. The optimum scale method was the most flexible and had the greatest reduction in the generalized variance. Using this method, relative importance was given to each endpoint. With a different objective criterion, the Gompertz and the logistic transformation method agreed that respiration was important. In addition to respiration, the Gompertz transformation increased the severity of piloerection while the motor activity was increased for the logistic. The Box-Cox transformation was the least flexible and emphasized sole importance of respiration. It indicated that any sign of respiration was detrimental. For this example given the categorical nature of all endpoints, the method of choice would be the optimum scale method.

3.5.5 Penalized Optimal Index

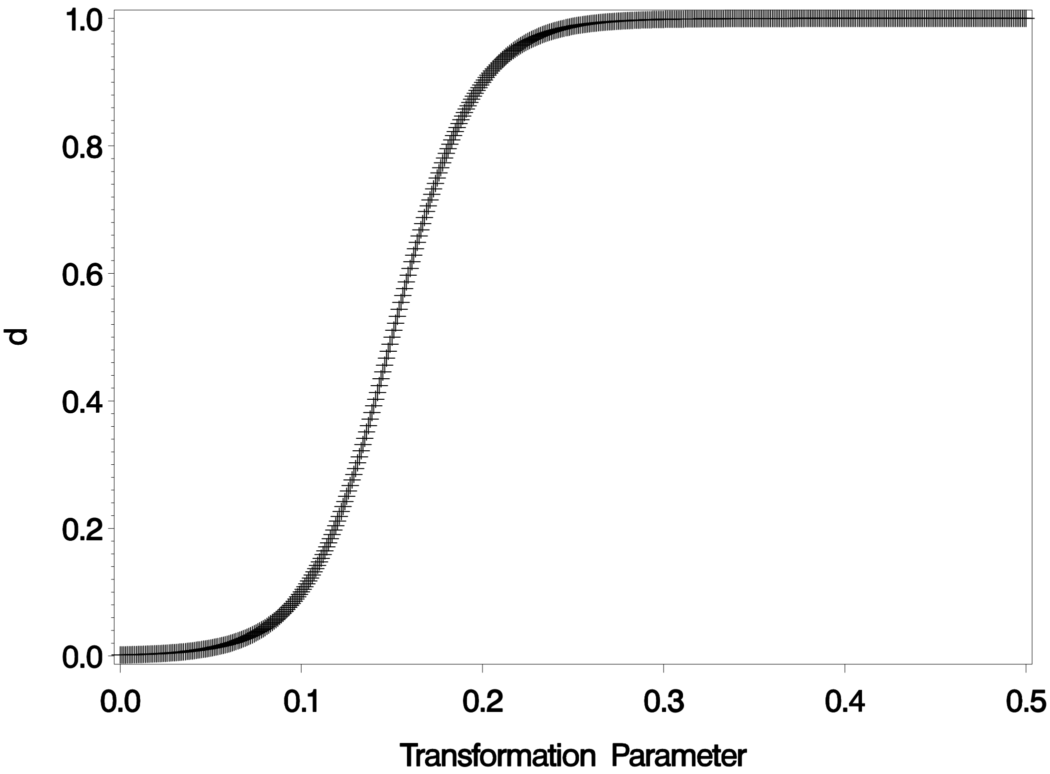
Recall the Box-Cox example given in section 3.5.4 where the optimal transformation parameters are listed in Table 3.6. Here, out of the four endpoints combined in the score, nearly all of the weight was focused on respiration. Although this index is statistically optimal for these data, the characteristics of such an index were not acceptable.

Suppose investigators desire to have information from all endpoints included in the score at some level. In this case, the penalized optimality can be used by including desirability functions defined to enforce certain criteria. Particularly, suppose investigators wish to have all weights to at least equal 0.1 with values of 0.2 or higher being preferred. This can be incorporated by using the bigger-is-better desirability function as described in eq (2.8). Specifically, this condition may be expressed as

$$d_i(\beta_{i(\min)}) = \left[1 + \exp\left(-\left(\frac{\beta_{i(\min)} - 0.15}{0.0228}\right)\right)\right]^{-1}$$

where $\beta_{i(\min)}$ is the minimum of the transformation parameters ($\min[\beta_1, \beta_2, \beta_3, \beta_4]$) from the Box-Cox transformation. Additional characteristics can be included however, for illustration purposes, only the constraints on the weights $(\beta_{i(\min)})$ are used. Here the overall desirability function for the penalty is expressed as $D_{(Penalty)i} = d_i(\beta_{i(\min)})$. The Nelder- Mead algorithm in SAS (version 9.1.3) is used to determine values of β_i that jointly minimize $H\left(D\left(f\left(\beta\right)\right);\zeta,\theta\right)$ and $\left(1 - D_{(Penalty)i}\right)$ given Λ. The minimum value of the optimality criterion defined $\min H\left(D\left(f\left(\beta\right)\right);\zeta,\theta\right)$=81.72. Penalized indices were found for multiples of the minimum value of the optimality criterion for values of Λ where the maximum $l = (1,10,50)$. Figure 3.8 graphically represents the responses the desirability and the scaled generalized variance for the l^{th} multiple. For values of $l < 0.2$, the criteria imposed on the weight transformation parameters were not satisfied. The value of $l = 0.253$ was chosen where the generalized variance was 102.36. The corresponding design is presented in Table (3.9). Here we see it cost a 21% increase in the generalized variance to incorporate the specified criteria.

Figure 3.7: Desirability Shape for the Penalty Function



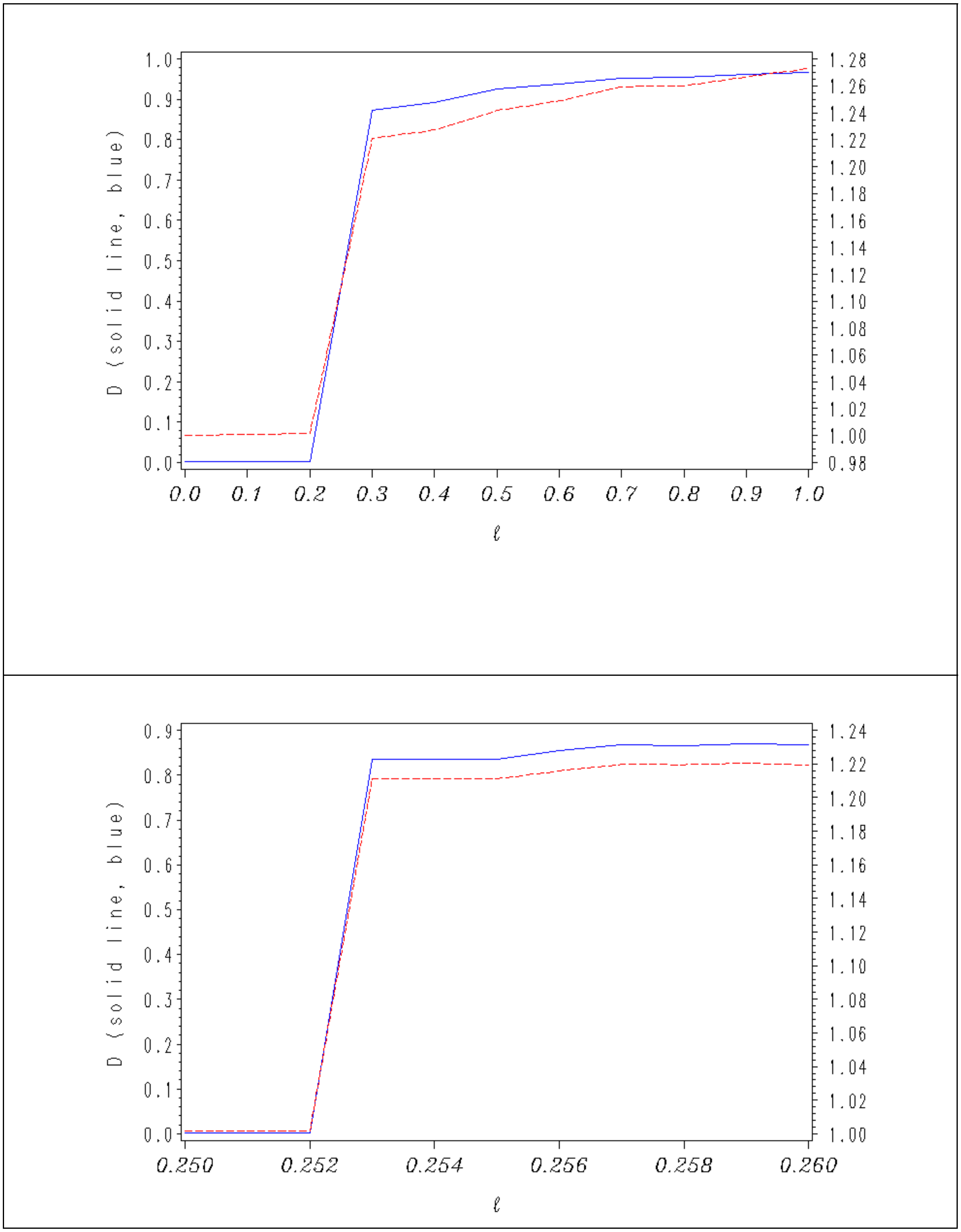


Figure 3.8: Penalized optimal responses using criterion given in (3.4)

Table: 3.9: Weight Parameters for Box-Cox Transformations

Optimization Type	β_1	β_2	β_3	β_4	Gvar
Optimal	0.0002	3.998	0.0006	0.001	81.718
Penalized Optimal	0.188	3.44	0.187	0.187	102.360

Table 3.10: Transformed Values for the Optimal and Penalized Optimal Cases

	Assigned Scores	**Optimal**	**Penalized Optimal**
d1 Motor Activity			
Normal	1	1	1
	0.9	1	0.98
Decreased	0.8	1	0.96
	0.65	1	0.92
Little or None	0.5	1	0.88
d2 Respiration			
Normal	1	1	1
	0.725	0.28	0.33
Tachypnea (thoracic)	0.45	0.04	0.06
	0.4	0.03	0.04
Tachypnea (abdominal)	0.35	0.02	0.03
	0.25	0	0.01
Dyspnea	0.15	0	0
d3 Muscle Tone			
Normal	1	1	1
	0.9	1	0.98
Moderate Loss	0.75	1	0.95
	0.575	1	0.9
Severe Loss	0.4	1	0.84
d4 Piloerection			
Not Present	1	1	1
	0.9	1	1
Present	0.8	1	1

3.6 Discussion

In this chapter, we present methodology to optimize transformation parameters in the calculation of a morbidity score. This morbidity score is created using desirability function methodology where a weighted score could be used to indicate levels of importance for the endpoints contained in the score. In the usual sense of such a weighted composite score, the weights are subjective. The methods displayed in this chapter minimize such subjectivity by using empirical techniques to optimize the parameters of the transformations functions used to create the composite score. Shapes/values for the desirability functions for each of the outcomes must be defined before the analysis is performed. Such shapes are best defined by a group of individuals knowledgeable about each outcome. In our case, shapes were defined in collaboration with the scientist who conducted the study. As noted by Coffey et al (2007), Since these desirability functions are developed *a priori* the approach can be standardized across studies and even across laboratories.

In this analysis, several complexities arose. The first challenge was the complexity of the data set itself. When using the Cox regression model, the presence of time-dependent covariates complicates the analysis and makes it harder to see the relationship. Secondly, the usual goodness of fit methods for model comparisons does not apply here because the independent variable changes for each model.

A limitation of the optimum scale method is that it can only be used with a limited number of categorical endpoints. Thus, when endpoints are continuous, other transformations need to be considered (e.g. logistic). Also further consideration needs to be placed upon the possibility of correlation between the endpoints that are aggregated in the composite score. Discrepancies in the results could be based on this problem

We demonstrated the use of penalized optimal methods to combine statistical optimality and expert opinion. Here desirability functions were used to penalize the designs and impose more desirable characteristics to the transformation parameters. In our example, the results of the design were either penalized or not. This may be due to having only one characteristic implemented in the desirability for the penalty. In this example, the single characteristic could have been imposed by adding a constraint however; the penalty function was used for general illustration purposes. Ideally, multiple properties would be imposed and combined in the penalty desirability. An example of this will be given in the next chapter.

CHAPTER 4

Development of a Severity Index for Pancreatitis

4.1 Introduction

Chronic pancreatitis (CP) has been defined as a continuing inflammatory disease of the pancreas characterized by irreversible morphologic changes that typically cause pain and/or permanent loss of function (Etemad and Whitcomb 2001). The natural history of specific morphologic stages of chronic pancreatitis is not well defined (Sandhu et al., 2007) and classification of its various forms are challenging. The most widely used classification systems are the Marseille classification system of 1963 with revisions made in 1984 and 1988 and the Cambridge classification of 1984 (Etemad and Whitcomb 2001). All of these classification systems distinguish between acute and chronic pancreatitis. The Cambridge classification uses imaging features to provide a grading severity system but it does not distinguish the different forms of CP on the basis of etiology and clinical outcome (Uomo 2002). The Marseille classification systems are said to be more useful in defining chronic pancreatitis than classifying it where the Cambridge system is useful as a staging system once chronic pancreatitis is diagnosed (Etemad and Whitcomb 2001). An ideal disease classification system for chronic pancreatitis would be simple, objective, accurate and relatively non-invasive incorporating etiology, pathogenesis, structure, function, and clinical status into one overall schema (Lankisch and Banks 1998).

In many clinical studies, it is common that researchers generally obtain measures on multiple endpoints that they believe are key to diagnosing chronic pancreatitis. Often, it is not clear which of these endpoints should be designated as of primary importance. Here we introduce using the desirability function approach as a way of combining multiple responses into a single unitless overall composite score. Desirability functions are widely used in engineering literature for product optimization and were introduced by Harrington (1965). In the toxicology literature, Coffey et al (2007) demonstrated the use of desirability functions to create a composite score comprised of multiple outcomes of various data types (continuous, ordinal, etc.) for toxicity dose-response experiments. This method combined with the use of a direct search procedure was used in the medical field to titrate dose combinations for individual patients (Shih 2003). Engineering literature has described by weighting individual components that the composite score can emphasize relative importance of certain outcomes (Derringer 1994). In the previous chapter we describe such weights to be transformations and find optimal transformations for each component.

In this Chapter we will demonstrate the use of desirability function methodology to create a composite of clinical outcomes for Chronic Pancreatitis (CP). Following the methods described in Chapter 3, we find the “optimal” transformation parameters of the composite score using the Box-Cox transformation of each component in the composite severity index. The approach is optimal in the sense that it finds the transformation parameters that maximize the determinant of the information matrix. As a result we in turn minimize the generalized variance of the parameter estimates for a pre-specified model.

Here we propose methods to combine multiple endpoints in an overall composite score. This score is easy to interpret and gives a quick overall assessment of health, toxicity, disease progression etc. In addition, we propose using optimization algorithms to guide us towards 'optimal' transformations if the individual desirability functions. When resulting transformations oppose the guidelines based on clinical expertise, we propose incorporating a penalized optimization similar to Parker and Gennings (2008).

Although full details of the general method are described in Chapter 3, a brief overview is provided in 4.2. An example is given where we demonstrate the use of the methodology in the development of a severity index score for pancreatitis. We then demonstrate how implementing a penalized optimality criterion can make some characteristics more appealing and still have relatively good statistical properties.

4.2 Motivating Example

The objective of this example is to demonstrate the use of the desirability function methodology in the development of a severity index score for pancreatitis. Moreover, we want to predict a patient's disease progression at least six months out given the value of the patient's severity index score at baseline. Disease progression in this analysis was defined as patients having any of the following: exocrine failure, endocrine failure or complications such as pseudocyst and bile duct stricture. An ordinal 'response' score was created which counted the number of these conditions for each patient. That is, Y=0 if none were present; Y=1 if only one was present; Y=2 if only two were present; Y=3 if only 3 were present and Y=4 if all four were present.

A total of eight-nine patients were seen at the pancreatitis center at Virginia Commonwealth University Medical Center for Chronic Pancreatitis and followed over time. Exocrine failure was defined by the presence of steatorrhea whereas endocrine failure is described by the presence of diabetes. Complications of Chronic Pancreatitis were defined clinically through magnetic resonance imaging (MRI). Exocrine failure is the inability to properly digest food due to the lack of digestive enzymes that are made by the pancreas. Endocrine failure is characterized by diabetes mellitus which is a condition where the pancreas does not produce enough insulin. Bile duct stricture is a narrowing or blockage of the bile duct and pseudocyst are a collection of fluids that may be a result of an injured duct. Having an increased number of these characteristics was defined as disease progression by the study investigators who are physicians at VCU medical center. Whether a patient has complications such as bile duct stricture or a pseudocyst was recorded at the time of the initial MRI as well as at a follow-up visit. If a patient was noted for having these complications at either the initial MRI or follow-up visit then they are indicated as having these complications (bile duct stricture or pseudocyst) in the follow-up score. To create the follow-up response score an analysis date of December 1, 2008 was chosen and any patient having exocrine failure or diabetes by this date is indicated in the follow-up score.

The average age of a patient was 48 years with a minimum and maximum age of 22 and 71, respectively. Of the 89 patients, 48 (54%) were males. The race of the study patients were categorized as Caucasian (45%) and other (55%). Descriptive statistics for the continuous variables are presented in Table 2 and the column proportions for each variable by response categories are listed Table 3.

Table 4.1: List of Structural and Behavioral Variables

Concomitant Alcohol Use (Y/N)	Pancreatic Atrophy (Y/N)	MPD Leak (Y/N)
Ongoing Smoking (Y/N)	MPD Calculi (Y/N)	SBE>3 (Y/N)
Contour Abnormality of the Bile Duct (Y/N)	MPD Stricture (Y/N)	MPD Size (mm)
MPD Irregularity (Y/N)	Side Branch Size (mm)	

Table 4.2: Descriptive Statistics for Continuous Variables

Variable	N	Mean (SD)	(MIN, MAX)
Side Branch Size (mm)			
0	22	1.7 (2.08)	(0, 8)
1	26	2.03 (1.53)	(0, 7.2)
2	31	2.7 (1.93)	(0, 8.3)
3	7	2.45 (2.07)	(0, 5.9)
4	1	3.8	3.8
MPD Size (mm)			
0	22	4.71 (4.25)	(1.6, 21)
1	26	4.7 (2.43)	(1.7, 9.2)
2	31	5.15 (3.27)	(1.2, 11.8)
3	7	7.27 (5.95)	(2, 18)
4	1	3.7	3.7

Table 4.3: Row Proportions of Each Variable by Number of Outcomes

		Column Proportions					
Variable		**0**	**1**	**2**	**3**	**4**	***Total Count***
Alcohol							
	No	0.61	0.69	0.61	0.25	1	54
	Yes	0.39	0.31	0.39	0.75	0	35
Smoking							
	No	0.17	0.35	0.36	0.13	0	25
	Yes	0.83	0.65	0.65	0.88	1	64
SBE>3							
	No	0.55	0.19	0.36	0.25	0	30
	Yes	0.46	0.81	0.65	0.75	1	58
MPD Stricture							
	No	0.82	0.73	0.68	0.86	0	64
	Yes	0.18	0.27	0.32	0.14	1	23
MPD Calculi							
	No	0.82	0.81	0.84	0.71	1	71
	Yes	0.18	0.19	0.16	0.29	0	16

Table 4.3 Continued: Column Proportions of Each Variable by Number of Outcomes

		Column Proportions					
Variable **MPD Leak**		**0**	**1**	**2**	**3**	**4**	***Total Count***
	No	1	0.89	0.94	0.88	1	82
	Yes	0	0.12	0.07	0.13	0	6
Pancreatic Atrophy							
	No	0.46	0.42	0.48	0.75	0	42
	Yes	0.55	0.58	0.52	0.25	1	46
Contour Abnormality							
	No	0.77	0.77	0.61	0.5	1	61
	Yes	0.23	0.23	0.39	0.5	0	27
MPD Irregularity							
	No	0.23	0.15	0.16	0.25	0	16
	Yes	0.77	0.85	0.84	0.75	1	72
Total Count		22	26	31	8	1	88

Given the ordinal nature of the response variable (Y=0, 1, 2, 3 or 4), a proportional odds model is used to determine if the number of outcomes is associated with a worsening severity index score. The model is of the form:

$$\text{logit}\left[P\left(Y \geq j \mid x\right)\right] = \log\left[\frac{P\left(Y \geq j \mid x\right)}{1-P\left(Y \geq j \mid x\right)}\right] = \alpha_j + \theta' x \quad \text{for } j = 0,1,\ldots,J-1 \quad (4.1)$$

where α_j are the intercepts for the j response categories and θ is the slope parameter. This model assumes that the odds ratio ($\exp(\theta)$) is constant for all categories. That is, it assumes the slope is the same for the categories only allowing for different intercepts. The general form of the likelihood for a sample of n independent observations is

$$L = \prod_{i=1}^{n}\left[\prod_{j=0}^{J} \pi_j(x_{ij})^{Y_{ij}}\right]$$

For J=4,

$$L = \prod_{i=1}^{n}\left[\ \pi_0(x_i)^{Y_{0i}} \bullet \pi_1(x_i)^{Y_{1i}} \bullet \pi_2(x_i)^{Y_{2i}} \bullet \pi_3(x_i)^{Y_{3i}} \bullet \pi_4(x_i)^{Y_{4i}}\ \right]$$

Thus it follows that the log-likelihood is given by

$$LL = \sum_{i=1}^{n}\left[\ Y_{0i}\ln\left[\pi_0(x_i)\right] + Y_{1i}\ln\left[\pi_1(x_i)\right] + Y_{2i}\ln\left[\pi_2(x_i)\right] + Y_{3i}\ln\left[\ \pi_3(x_i)\right] + Y_{4i}\ln\ \left[\pi_4(x_i)\right]\right]$$

where the component probabilities of the log-likelihood are given by

$$\begin{aligned}
\pi_4 &= P\left(Y \geq 4\right) = P\left(Y = 4\right) \\
\pi_3 &= P\left(Y \geq 3\right) - P\left(Y \geq 4\right) = P\left(Y = 3\right) \\
\pi_2 &= P\left(Y \geq 2\right) - P\left(Y \geq 3\right) = P\left(Y = 2\right) \\
\pi_1 &= P\left(Y \geq 1\right) - P\left(Y \geq 2\right) = P\left(Y = 1\right) \\
\pi_0 &= 1 - P\left(Y \geq 1\right) = P\left(Y = 0\right)
\end{aligned}$$

where,

$$\pi_4 = \frac{\exp(\alpha_4 + \theta' x_i)}{1 + \exp(\alpha_4 + \theta' x_i)}$$

$$\pi_3 = \frac{\exp(\alpha_3 + \theta' x_i)}{1 + \exp(\alpha_3 + \theta' x_i)} - \frac{\exp(\alpha_4 + \theta' x_i)}{1 + \exp(\alpha_4 + \theta' x_i)}$$

$$\pi_2 = \frac{\exp(\alpha_2 + \theta' x_i)}{1 + \exp(\alpha_2 + \theta' x_i)} - \frac{\exp(\alpha_3 + \theta' x_i)}{1 + \exp(\alpha_3 + \theta' x_i)}$$

$$\pi_1 = \frac{\exp(\alpha_1 + \theta' x_i)}{1 + \exp(\alpha_1 + \theta' x_i)} - \frac{\exp(\alpha_2 + \theta' x_i)}{1 + \exp(\alpha_2 + \theta' x_i)}$$

$$\pi_0 = 1 - \frac{\exp(\alpha_1 + \theta' x_i)}{1 + \exp(\alpha_1 + \theta' x_i)}.$$

We obtain the maximum likelihood estimates of the parameters by differentiating the log-likelihood with respect to each of the parameters setting each of the equations equal to zero and solving for $\widehat{\theta}$. In this study, 11 possible anatomical and behavioral variables as listed in Table 1 were under consideration to combine into the overall severity index. The relationships of the ordinal response variable with all variables were studied independently using the proportional odds model.

To investigate a more complex multivariable relationship, a stepwise ordinal logistic regression was preformed in SAS version 9.1.3. A stepwise procedure utilizes the likelihood ratio test to determine which variables to include or exclude from the model. The significance level for a variable to enter the model was set at 40% and set at 50% to stay in the model. This criterion was loosely set to allow variables without requiring a strong association. The endpoints were then transformed into desirability values. Binary variables were assigned desirability values of 1 when endpoint response is “No” and 0.5 for the response of “Yes”. Desirability values for the continuous variables are transformed using the logistic CDF (2.9).

The severity score was then created by first aggregating the individual desirabilities of the endpoints that was significant in the stepwise logistic model. The severity score was then studied to investigate if the score is predictive of disease progression or worsening outcomes. Ideally, the more outcomes that comprise the score, the more generalizable the index will be. For this purpose, additional variables were added to the severity index one at a time in the order corresponding to the strength of association of the individual logistic regression. The relationship of the final severity index with disease progression was analyzed.

To find transformation parameters (β_j) that are optimally related to some external response variable (ζ) the methods of section 3.3 were applied. The optimal severity index was defined as $D_{(opt)i} = \left(f_1 \cdot f_2 \cdots f_k \right)^{1/k}$ for k=9 endpoints, where $f_j = f_j\left(d_j\left(y\right)\right) = d_{ij}^{\beta_j}$, i.e. the Box-Cox transformations. The objective function, $H\left(D\left(f\left(\beta\right)\right);\zeta,\theta\right)$, to be minimized by the Nelder-Mead algorithm was the generalized variance which is defined as the determinant of the covariance matrix for the estimated parameters in the ordinal logistic model. Following the optimal transformation methods in Chapter 3, a nonlinear optimization subroutine was used for parameter estimation in conjunction with the Nelder-Mead direct search algorithm (in SAS version 9.1.3) to find the transformation parameters that minimize the generalized variance of the proportional odds model.

To combine information from the empirical optimization with clinical expertise, we implement penalized optimality methods as described in section 3.4. Here a penalty function is used to penalize transformations to accommodate expert opinion. The penalty function is defined as $\left(1 - D_{(Penalty)i}\right)$ where $D_{(Penalty)i} = \left(d_{i1} \times d_{i2} \times \ldots \times d_{iq}\right)^{1/q}$ for the q characteristics preferred by the experts. Here we find transformations that jointly minimize the objective criterion $H\left(D\left(f\left(\beta\right)\right);\zeta,\theta\right)$ and the penalty function $\left(1 - D_{(Penalty)i}\right)$, similar to the strategy described by Parker and Gennings (2008).

4.3 Results

The results for the univariate analysis relating each demographical, anatomical and behavioral variable with the ordinal response score for disease progression are given in Tables 4.4 and 4.5. Only one variable (Side Branch Size (mm)) was independently positively associated ($\theta = 0.228,\ p = 0.034$) with the ordinal response variable not correcting for multiple testing. That is, for an increasing value of side branch size, patients are seemingly more likely to be in a higher response category (have more outcomes related to progression of pancreatitis). However, we suspect that a more complex multivariable relationship may exist; thus, a stepwise ordinal logistic regression was performed (SAS version 9.1.3).

Table 4.4: Independent Ordinal Regression for Demographic Variables

Model	***Variable***	***Estimate***	***SE***	***P-Value***
1	**Age**	0.00911	0.0182	0.6175
2	**Gender**	-0.1693	0.3863	0.6612
3	**Race**	0.0620	0.3851	0.8722

Table 4.5: Independent Ordinal Regression for Anatomical and Behavioral Variables

Model	*Variable*	*Estimate*	*SE*	*P-Value*
4	**Concomitant Alcohol Use**	0.4025	0.3954	0.3087
5	**Ongoing Smoking**	-0.1965	0.4277	0.6459
6	**SBE >3**	0.5897	0.4117	0.1521
7	**MPD Stricture**	0.4069	0.4440	0.3594
8	**MPD Calculi**	0.0323	0.5022	0.9487
9	**MPD Leak**	0.6054	0.7742	0.4342
10	**Pancreatic Atrophy**	-0.3340	0.3882	0.3897
11	**Contour Abnormality**	0.7001	0.4266	0.1008
12	**Side Branch Size**	0.2276	0.1071	0.0335
13	**MPD Size**	0.0822	0.0559	0.1415
14	**MPD Irregularity**	0.1520	0.5008	0.7615

Candidate variables for the severity score were those variables selected through the stepwise process. Using the desirability methodology, a severity score index was defined using concomitant alcohol use (d_1), side branch size (d_2), MPD Stricture (d_3) and MPD leak (d_4) as

$$D = (d_1 \times d_2 \times d_3 \times d_4)^{\frac{1}{4}}$$

Individual indices d_1, d_3 and d_4 are all assigned values of 1 if present and 0.5 if not present. The continuous variable side branch size is transformed to d_2 using a decreasing logistic CDF as describe in the smaller-is-better case (2.9) with the shape determined in collaboration with study investigators (Figure 1). Additional parameters with positive estimates are added to the severity score individually and the strength of the association is studied after each addition. The final severity score has the addition of the variables contour abnormality of the bile duct$\left(d_5\right)$, SBE >3 $\left(d_6\right)$, MPD Irregularity $\left(d_7\right)$, MPD Calculi $\left(d_8\right)$, MPD Size $\left(d_9\right)$ and is of the form

$$D = (d_1 \times d_2 \times d_3 \times d_4 \times d_5 \times d_6 \times d_7 \times d_8 \times d_9)^{\frac{1}{9}} .$$

MPD Size was transformed into $\left(d_9\right)$using a targeted desirability function and is shown in Figure (2).

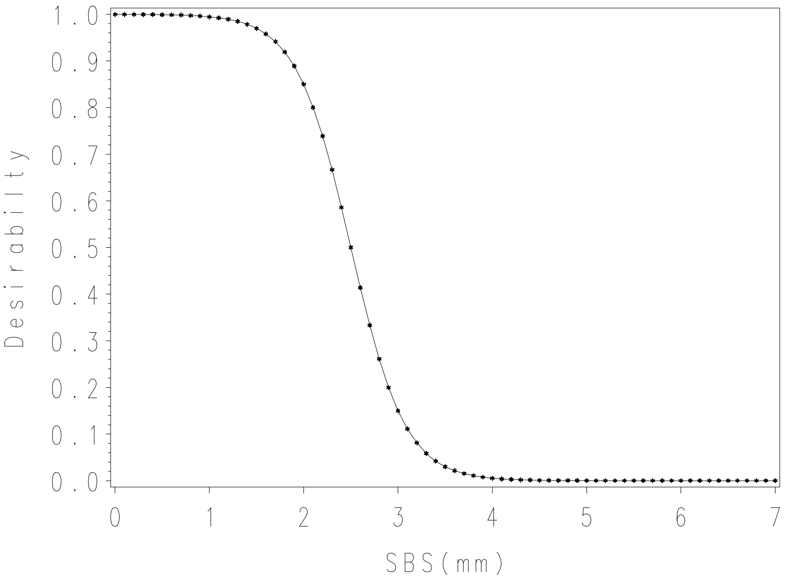


Figure 4.1: Individual desirability for Continuous Variable Side Branch Size (SBS)

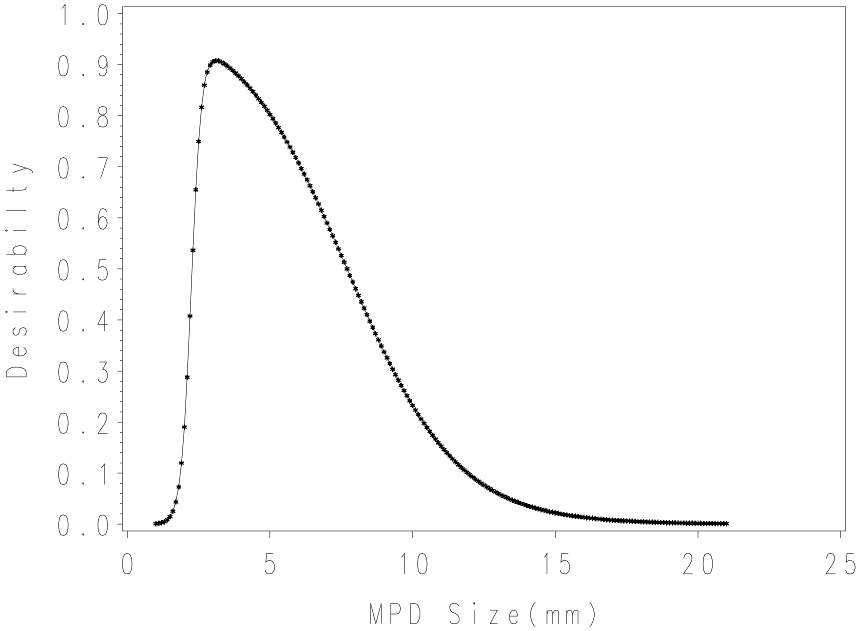


Figure 4.2: Individual desirability for Continuous Variable Main Pancreatic Duct (MPD) Size

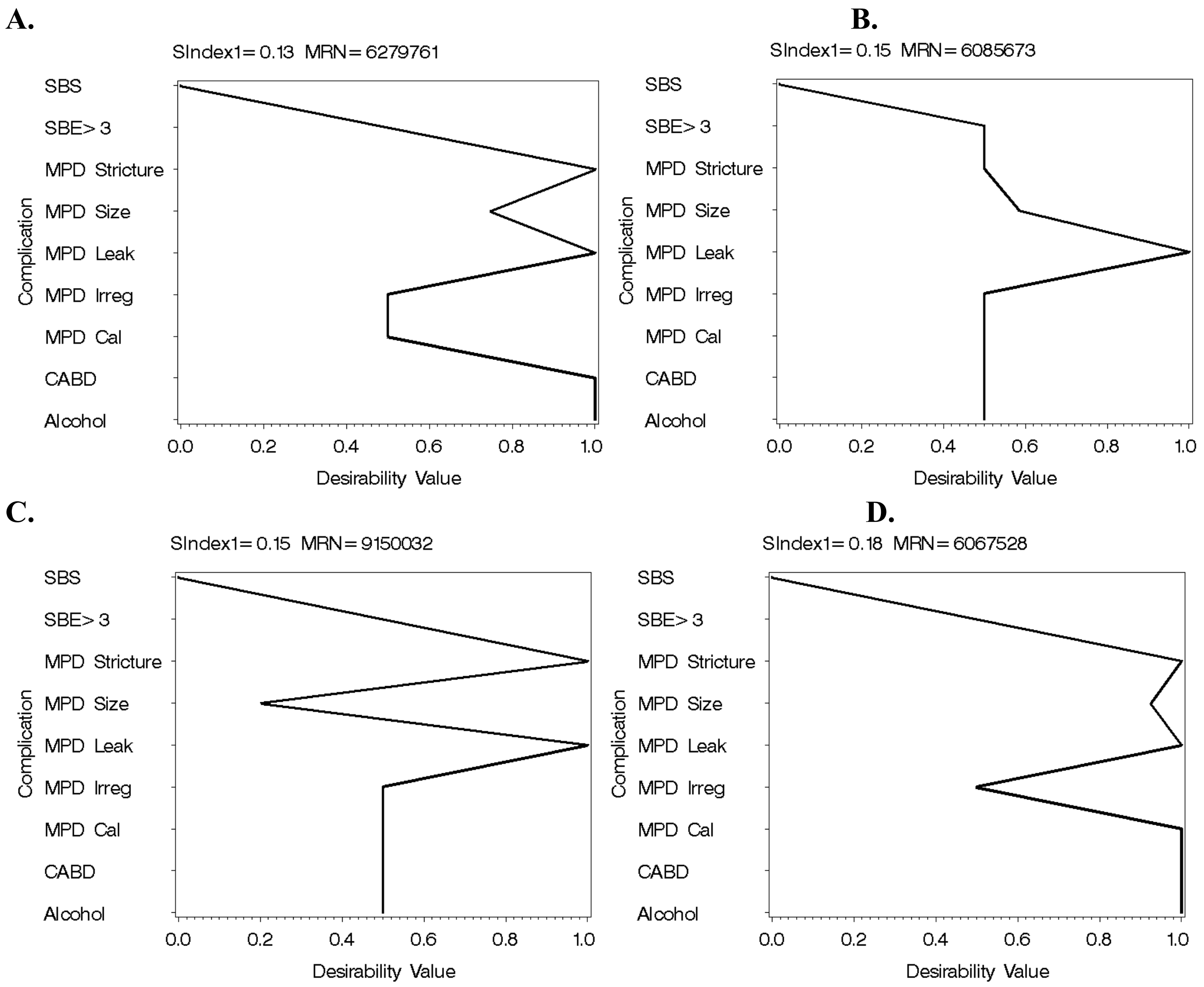
The ordinal logistic regression model in (4.1) for disease progression was parameterized to include the severity score, D. Maximum likelihood estimates of the unknown model parameters were found using the Fisher scoring algorithm (Proc logistic, SAS version 9.1.3). The score test for the proportional odds assumption of the ordinal regression model is satisfied $\chi^2(3) = 0.51$, $p = 0.92$. The severity index is negatively associated ($\theta = -2.10,\ p = 0.017$) with the ordinal response variable for disease progression (Table 4.6). That is as the severity index decreases, a patient is more likely to have multiple outcomes. For a one unit decrease (defined as 0.10) in the severity index the odds of moving to the next category is increased 1.23 times. The 95% Wald Confidence Interval for the odds ratio is (1.038, 1.467). Thus, a decrease of 0.1 in the severity index is associated with an increase in the odds of disease progression of 23%. Of the 89 patients in the study, 30 patients actually progressed (increased the # of outcomes related to pancreatitis severity). Of the 30 patients that progressed, 19 (0.63) patients had a value of the severity index less than the median (0.71). In this analysis, 32/59 (.54) of the patients that did not progress had a severity index that the greater than the median value.

Table 4.6: Parameter Estimates for the Ordinal Logistic Regression

Parameter	**Estimate**	**Standard Error**	**P-value**
Intercept 4	-3.20	1.12	0.004
Intercept 3	-1.02	0.63	0.107
Intercept 2	1.16	0.62	0.060
Intercept 1	2.51	0.66	<0.001
Severity Index	-2.10	0.88	0.017

Below in Figures 4.3-4.5, we show how a patient with a similar severity index score can have different complications. For illustration purposes we give plots for low (Figure 4.3 A-D), medium (Figure 4.4 A-D) and high (Figure 4.5 A-D) Severity index scores. This shows that regardless of the set of complication, the Severity index is a uni-dimensional comprehensive index score that gives an overall since of chronic pancreatic severity.

Figure 4.3: Profile plots for patients with severity index=0.13-0.18.



A. SIndex1= 0.13 MRN= 6279761

B. SIndex1= 0.15 MRN= 6085673

C. SIndex1= 0.15 MRN= 9150032

D. SIndex1= 0.18 MRN= 6067528

Figure 4.4: Profile plots for patients with severity index=0.54-0.58

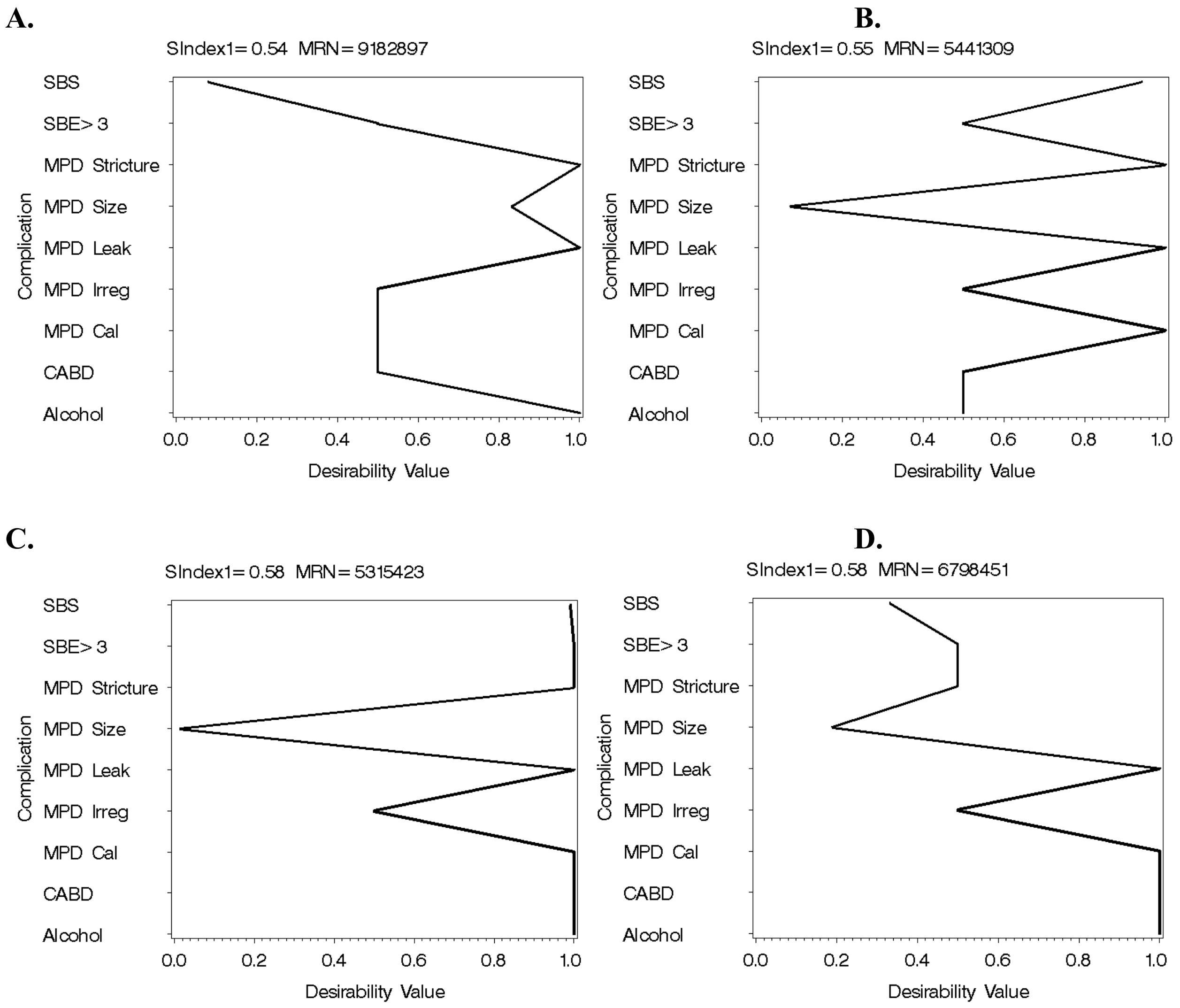
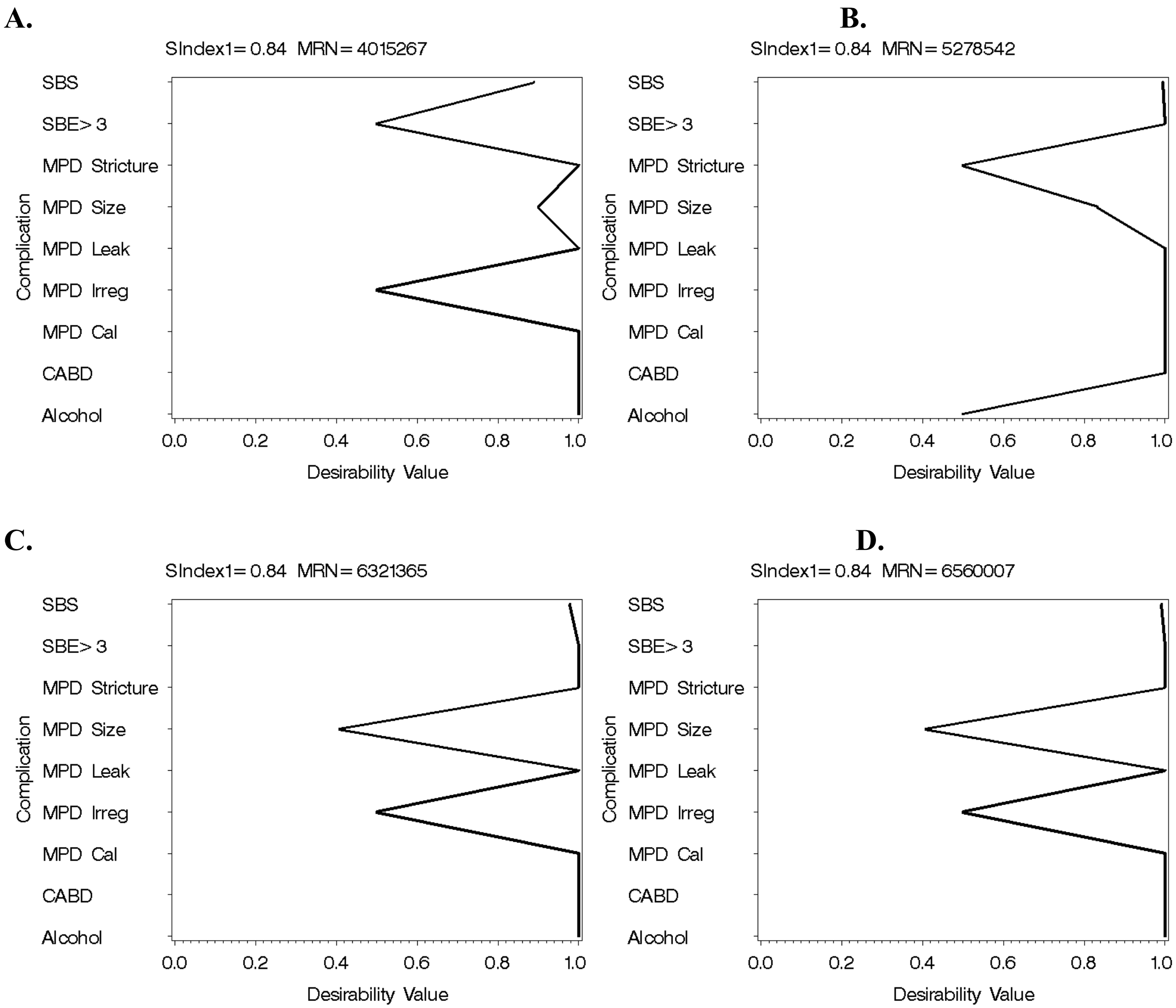


Figure 4.5: Profile plots for patients with severity index=0.84



The optimization algorithm as described in detail in chapter 3 was used to find the optimal Box-Cox transformation parameters. In this case we found the transformation parameters that minimized the generalized variance of the proportional odds model. The method was implemented using the Fisher scoring algorithm embedded within the Nelder-Mead algorithm over a grid of starting values for the parameters in SAS version 9.1.3. The optimized parameters are given in Table 7. Out of the 9 variables that comprised the index, relative importance was focused only on side branch size and MPD stricture. The most emphasis was placed on side branch size whose transformation parameter was 8.25 with the sum of all parameters constrained to equal 9, the number of components in the index. All other variables were essentially eliminated with transformation parameters close to 0. The optimized severity index was analyzed to study its relationship with the ordinal response score (Table 8). The score test for the proportional odds assumption was satisfied $\chi^2(3) = 2.99$, $p = 0.39$. The optimized severity index was significant ($\theta = -1.36,\ p = 0.008$). As the severity Index decreases, patients are more likely to have more pancreatitis severity outcomes. The corresponding odds ratio and 95% CI is 0.256 (0.796, 0.968).

Table 4.7: Optimal Box-Cox Transformation Parameters for the Severity Index

Endpoint	Optimal Transformation Parameter (β_i)
Side Branch Size	8.25
MPD Leak	0.01
MPD Stricture	0.62
Alcohol	0.02
CABD	0.02
SBE>3	0.02
MPD Irregularity	0
MPD Calculi	0.06
MPD Size	0

Table 4.8: Parameter Estimates for the Ordinal Logistic Regression with Opt. Parameters

Parameter	**Estimate**	**Standard Error**	**P-value**
Intercept 4	-3.73	1.03	0.0003
Intercept 3	-1.55	0.46	0.0007
Intercept 2	0.62	0.40	0.117
Intercept 1	1.99	0.44	<0.001
Severity Index	-1.30	0.50	0.009

Previously it was shown that the optimal severity index score put emphasis only on two parameters and essentially eliminated the others (Table 4.7). For generalizability, it is ideal to have more components that comprise the Index. Penalized optimality was implemented following the methods of Parker and Gennings (2008) to combine information from the statistical optimization with expert judgment. A SAS macro was used to find the find the optimal transformation parameters under penalized criteria using the Nelder-Mead Simplex Algorithm. Moreover, suppose investigators desire to have all transformation parameters equal at least 0.15 but not greater than 6.5. With these characteristics, the desirability function was created as described in section 2.4 where d_1 is defined as:

$$d_1 = \left[1 + \exp\left(-\left(\frac{Y^{(1)}{}_i - a^{(1)}{}_i}{b^{(1)}{}_i}\right)\right)\right]^{-1}$$

for $a^{(1)}{}_i = \left[\frac{Y_i^{*1} + Y_{i*1}}{2}\right]$,$b^{(1)}{}_i = \left[\frac{Y_i^{*(1)} - Y_{i*}{}^{(1)}}{2\ln\left(\frac{1-\gamma^{(1)}{}_i}{\gamma^{(1)}{}_i}\right)}\right]$ where the lower bound $Y_{i*}{}^{(1)} = 0.1$ and upper bound $Y_i^{*(1)} = 0.2$ for $\gamma^{(1)} = 0.1$. The smaller the better function for the given characteristics not to exceed 6.5 was created by

$$d_2 = \left[1 + \exp\left(\frac{Y_i^{(2)} - a_i^{(2)}}{b_i^{(2)}}\right)\right]^{-1}$$

for $a_i^{(2)} = \left[\frac{Y_i^{*(2)} + Y_{i*}{}^{(2)}}{2}\right]$,$b_i^{(2)} = \left[\frac{Y_i^{*(2)} - Y_{i*}{}^{(2)}}{2\ln\left(\frac{1-\gamma_i^{(2)}}{\gamma_i^{(2)}}\right)}\right]$ where the lower bound $Y_{i*}{}^{(2)} = 5.5$ and upper bound $Y_i^{*(2)} = 7.5$ for $\gamma^{(2)} = 0.1$. The desirability function for the desired properties of the transformation parameters was expressed as $D_{(Penalty)i} = \left(d_1 \bullet d_2\right)^{1/2}$.

Penalized transformation values were calculated for the scale factor λ values ranging from 0 to10. The starting values for the transformation parameters were set at the starting values in which the minimized generalized variance was found. The desirability and generalized variance was plotted against λ. The range of λ was then increased or decreased depending on the plots. For a given λ value, if there was a spike in the value of the generalized variance then this value was further investigated. We set the penalty parameter λ at a specific value, and searched over a grid of starting values for the transformation parameters. The search resulted in the smallest generalized variance being found over a grid of values for each λ. For the end result, we chose λ such that we have increased desirability defined by the penalty function and the corresponding increase in the generalized variance is tolerable and reasonable. The plot of the achieved desirability and generalized variance are given in Figures 4.6 and 4.7. The chosen λ for the optimal penalized transformation was that of λ=0.1. Here the desirability value was 0.75 with an increase of the generalized variance of approximately 16%. Comparisons of the three model transformation types are listed in Table 4.9. As expected, the optimal transformation case was the most significant followed by the penalized transformation case, then the un-weighted case. However, we saw that the model prediction for the optimal case and the penalized case are essentially the same. In Table 4.10, the optimal transformation parameters and the penalized optimal parameters were presented. Approximately 95% of the weights were attributed to side branch size and MPD stricture in the optimal transformation case. In the penalized transformation case, the weights were somewhat more distributed across all nine variables.

Table 4.9: Parameter Estimates of the Different Severity Indices

Type	Index Estimate	P-Value	Odds Ratio
Un-weighted	-2.10	0.02	0.81
Optimal Transformation	-1.30	0.009	0.88
Penalized Transformation	-1.35	0.01	0.87

Table 4.10: Transformation Parameters for Optimal and Penalized Cases

	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	*GVar	Desirability
Optimal	8.25	0.013	0.623	0.016	0.015	0.021	0.0002	0.060	0.0003	$6.6\text{x}10^{-5}$	0.037
Penalized	5.67	0.52	0.16	0.16	0.32	0.16	0.16	1.69	0.17	$7.7\text{x}10^{-5}$	0.77

*GVar is the generalized variance of the estimated proportional odds model

Figure 4.6:Desirability and Scaled Generalized Variance vs. λ Depicted in the Full Range

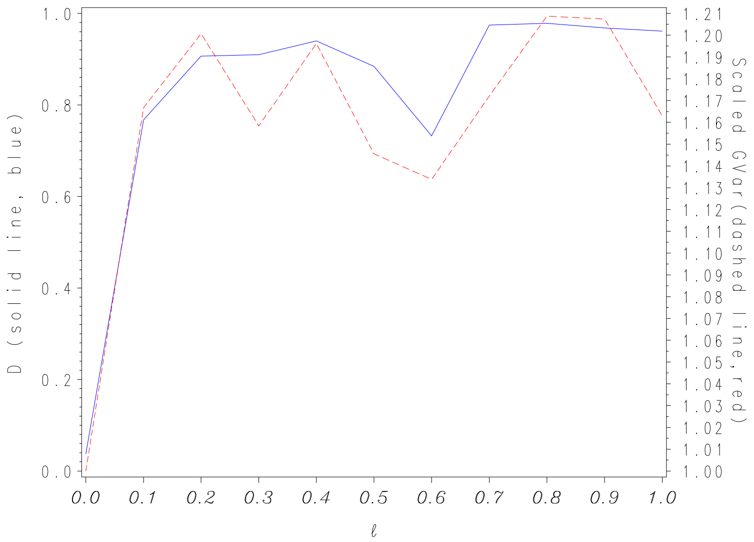


Figure 4.7: Desirability and Scaled Generalized Variance vs. λ Depicted in the Restricted Range

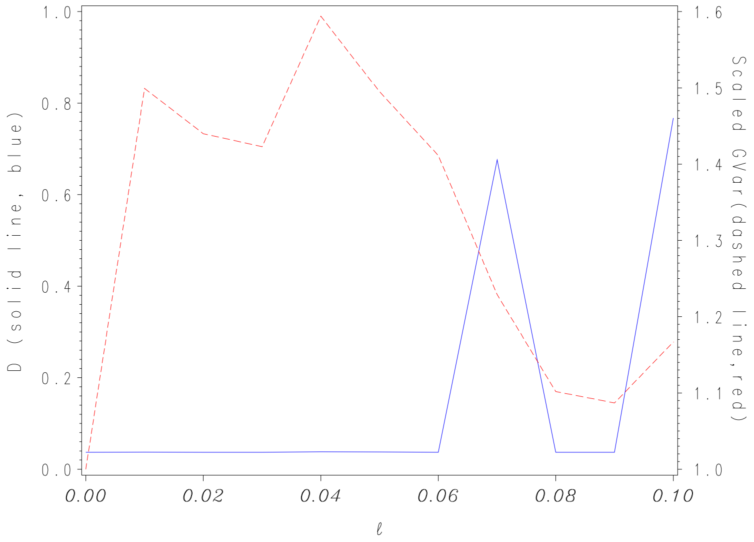


Table 4.11: Disease Progression vs. Median Penalized Optimal Severity Index

	Disease Progression		
Severity Index	**Yes**	**No**	**Total**
< Median	18	23	41
> Median	12	36	48
Total	30	59	89

For the penalized optimal score, of the patients that actually progressed, 60% of the patients had a lower (less than the median value=0.85) severity index score (See Table 4.11). For the patients that did not progress, 61% of them had a severity index that was greater than the median value. This is analogous to the sensitivity/ specificity respectively of a test. However, in this analysis, we are using the index score to predict disease progression in the future where sensitivity and specificity are defined on current disease status.

In this Chapter there were eleven anatomical or behavior variables that are believed to be associated with severity of pancreatitis and possibly predict disease progression. We have demonstrated methodology to comprise as many of the variables as possible into a Severity Index. Because of the ordinal nature of the response variable, a proportional odds model was used. Initially, the Index was created using only the four variables that were found to be significant in a stepwise procedure. Additional variables were added to the Index one at a time and the ability of the index score to predict the ordinal response was examined after each addition. In each case the index was significantly associated with disease progression. Once the final Index was created the goal was then to find the Box-Cox transformation parameters that minimized the generalized variance of the given proportional odds model. Such transformation parameters are found using an optimization algorithm that is comprised of a nonlinear optimization subroutine and the Nelder Mead direct search algorithm. Although the parameters found satisfy a statistical optimality criterion, there may be practical issues concerning the characteristics. Out of the 9 variables that were included in the index, relative importance was weighted only on 2 variables and the effects of the others were essentially eliminated.

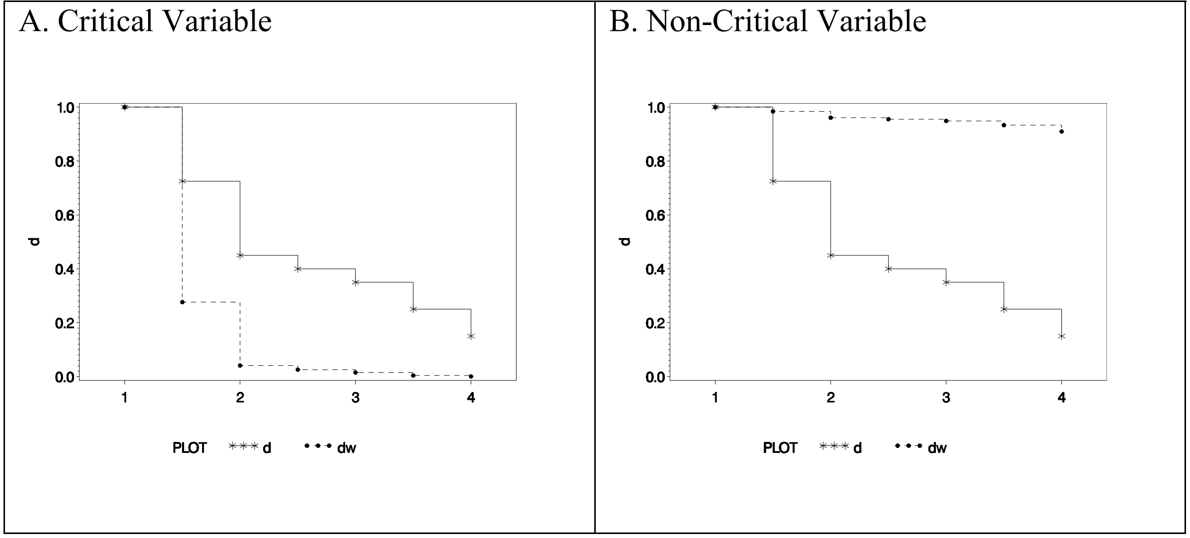
These results motivated the use of penalized optimal methods to combine the statistical optimality with expert opinions from investigators. Desirability functions were used to penalize and impose more desirable characteristics on the transformation parameters For illustration purposes, mild constraints were used to penalize the cases where transformation parameters were not at least 0.15 or if the parameter exceeded a value of 6.5. The penalized criterion seeks to jointly minimize the generalized variance and imposed penalty function. A penalized case was chosen such that there was a sizable increase in desirability along with a tolerable increase to the generalized variance of the proportional odds model. It was expected the penalized parameters would be associated with a smaller generalized variance than that of the original score but not as small as that of the optimal case. Although the transformation parameters were different for the optimal and penalized cases (Table 4.10), it was shown that the model predictions were essentially the same.

4.4 Discussion

In this dissertation, we combine multiple endpoints into a single composite score using desirability functions. In this chapter, a severity index (composite score) was calculated where we used the data to determine the relative importance of each component that comprise the index. Specifically, the severity index was derived from multiple baseline MRI measurements and a behavioral variable (alcohol usage) where the measured properties were transformed to the desirability scale, i.e. zero, least desirable, to one, most desirable. The MRI measurements were all structural features of the pancreas (i.e. main pancreatic duct irregularity, side branch size etc.) that characterize chronic pancreatitis.

The relative importance of each of these features can be taken into account by weighting each of the components of the index (Derringer 1994). However, how important a feature is relative to another feature may be difficult for investigators to quantify. Our method empirically determined these values. The weight variable transformed the individual desirability functions to indicate how critical a variable may be relative to others (Harrington 1964). In chapter 3 we demonstrated the effect of the weighting/transformation parameter w_j with figure 4.8. The empirical value for w_j resulted in steepening the transformation curve for critical variables (Figure 4.8 A: denoted curve dw) and for less critical variables transformation curves are nearly horizontal (Figure 4.8 B: denoted by curve dw).

Figure 4.8: Hypothetical example of transformation effects



A. Critical Variable

B. Non-Critical Variable

In this chapter, the outcome of interest was disease progression, where disease progression was defined as the patient having endocrine failure, exocrine failure or complications such as having a pseudocyst or bile duct stricture. A more clearly defined outcome such as mortality would be ideal to use to determine a severity score that could identify patients most at risk of dying. However, determining a severity score predictive of disease progression allows earlier detection of chronic pancreatitis. There are several benefits for creating a composite score. One benefit is the advantage of dimension reduction. In this example, we took many variables from MRI measurements and combined the information into an overall severity index (i.e., changing from p-dimensions to one-dimension). This index is an overall assessment of pancreatitis severity where a patient with a score close to zero indicates a severe condition and a score close to one is most desirable. Even though many variables may be used to describe severity, by combining the variables into a single composite score we have reduced the information into a single dimension. Such a score ranks patients at baseline based on their disease severity from most severe to least severe. Composite scores are easily interpreted. Examples of composite scores that are commonly used in practice include the APACHE score for intensive care unit patients and the MELD score for liver transplant patients.

It is not necessarily true that disease progression is linearly related to the severity index. It may be helpful to determine a threshold in the severity index beyond which the probability of Chronic Pancreatitis progressing increases. The model could be re-parameterized to be a piecewise threshold model where the threshold is estimated from the data. If a patient has a score beyond the threshold, the physician could decide to treat the patient more aggressively; a patient with a score above the threshold could continue to be monitored.

A concern that may arise could be the number of variables that are aggregated in the score. In practice, one would want to perform as few invasive procedures as possible. In this case, many of the variables that comprise the index score are from MRI values of the pancreas where information on many structural features was measured. Since the information was readily available from the MRI, and physicians felt that all features were important, the initial strategy was to use many variables for generalizability. In cases where tests may be invasive or expensive, one could find the best subset that would be indicative of a specific condition. After applying the proposed optimization methodology, the end result was a parsimonious model where relative importance was only emphasized on two variables: side branch size and MPD stricture. MPD stricture may be a surrogate for some of the other variables considered: MPD irregularity, MPD calculi and MPD size. The size of the side branch provides more information than just an indication of whether the side branch is greater than 3mm (SBE>3). In the case where the results of the statistical optimality is not practical, we suggest using a penalized optimization where we are able to combine statistical optimality with characteristics that physicians find important.

Ideally, to examine the reliability of the severity index, a cross validation could be performed where the data would be split into two disjoint sets. This would allow us to examine the similarities of the optimized weighting components and also test the model prediction. However, in this example, our sample size of 89 was not large enough to implement these methods. In this dissertation, we did not study the impact of sample size.

The challenge of getting physicians to accept the proposed methodology in this dissertation may best be addressed in phases. An important initial phase is to publish the methodology in a statistical journal. It is hard to break common practice so we must first show that it is a valid approach. Once the methodology is accepted in a statistical journal, we would focus on the application of the methods and publish in journals specific to certain subject-matter. It is also a goal to continue working with active researchers who are willing to calculate the index on their patients and study how well the index works over an extended period of time. For a particular subject-matter, it may be beneficial to consult with multiple physicians from different hospitals for the initial creation of the composite score. This may reduce bias and gain more initial information on certain features. In the end, if the physicians see the benefit for efficiency, predictability and ease of interpretation, they may be more likely to continue to use the methodology.

Chapter 5

Summary Remarks and Future Work

5.1 Summary

This work has introduced methodology for creating a composite score that combines a set of multiple response variables and optimally linking the score to an external outcome variable through an objective function of interest. These responses can be of different types (binary, count, ordinal etc.) and were aggregated using desirability functions. In Chapter 2, we presented a literature review of multi-response optimization. In this section, details of the desirability function methodology are described. Here we also illustrate the use of desirability functions in different literature.

In Chapter 3, we presented methodology for creating the optimized composite score. The method is implemented using a pre-clinical example where we develop a morbidity composite score that is related to the instantaneous hazard of death. In this example multiple biological and behavioral responses were combined into a single morbidity composite score. Several transformations are considered and using the Nelder-Meade direct search algorithm, optimal transformation parameters were found using the proposed method. Such parameters are optimal in the sense that we found the transformation parameters that minimized/maximized some given objective criterion. Because statistically optimal transformations found may not be in agreement with expert judgment, a penalized optimality criterion was implemented. Using such a criterion allows us to combine expertise with statistical optimality.

In Chapter 4, the methodology is implemented using data from a clinical study. Here we developed a severity index which can be used to track disease progression and predict worsening conditions using variables describing patient behavior and physiological measurements from MRI. For this analysis, “worsening conditions” were defined as patients having any number of complications (exocrine failure, diabetes, pseudocyst and bile duct stricture) six months from baseline. For a given patient, the number of complications was summed and an ordinal response score was created. The Box-Cox transformation was used and optimal transformation parameters were found using the Nelder-Mead direct search algorithm to minimize the generalized variance of the regression parameters. Similarly to Chapter 3, a penalized optimality criteria was implemented to combine clinical expertise with statistical optimality.

5.2 Limitations and Future Work

A limitation of the methods described in this dissertation is that it is dependent upon the data set. That is, the use of the method is only as good as the data. For instance, let’s consider the data used in the example given in Chapter 3. In this study, the animals were examined twice a day for five days and morbidity scores were calculated. In this case, the majority of the animals either died very early in the study or they survived. Therefore we were not able to gain information from responses that could have occurred at in between time points.

In this method, we combine multiple endpoints into a single composite score that is an overall gestalt of information. In many studies, some of these endpoints could be correlated which could possibly have an effect on the optimizations process. In future work, we would like to address this concern by taking into account the correlated variables. In this case, do we need to change the structure of the desirability function to address the correlation? We may also want to do a simulation to examine if there is an effect on adding too many variables to the score. Will this dilute the importance of other variables in the score? If this is the case, what number would be considered "too many".

In this dissertation, when using the Box-Cox transformation, we added a constraint where the sum of the transformation parameters would equal the total number of variables that are in the index score. Another suggestion would be to normalize this variable where the sum would equal to 1. Here, regardless of the number of endpoints combined, the sum of the transformation parameters would always be 1.

As medical practice advance, we would need to update the index. For example, the endpoints that medical doctors deem important for diagnosing chronic pancreatitis today may change in the future. In this case we would need to update the variables that comprise the index score to be in line with what doctors examine in practice.

List of References

List of References

Bull, S. B. (1998). Regression models for multiple outcomes in large epidemiologic studies. *Statistics in Medicine* ,**17,** 2179-2197.

Centers for Disease Control and Prevention(CDC). Emergency Preparedness and Response: Bioterrorism Agents. Retrieved, November 2007 Available at: http://www.bt.cdc.gov/agent/agentlist-category.asp

Coffey, T., Gennings, C., Moser, V.C. (2007). The simultaneous analysis of discrete and continuous outcomes in a dose–response study: Using desirability functions. *Regulatory Toxicology and Pharmacology,* **48**, 51–58.

Collett, D. (2003) *Modeling Survival Data in Medical Research* (2nd ed). London: Chapman & Hall

Clain, J E. (1999). Diagnosis of chronic pancreatitis is a gold standard necessary?. The Surgical clinics of North America, **79**(4), 829.

Creason, J P. (1989). Data evaluation and statistical analysis of functional observational battery data using a linear models approach. International journal of toxicology, **8**(1), 157.

Derringer, G., Suich, R. (1980). Simultaneous optimization of several response variables. *Journal of Quality Technology* **25,** 199-204.

Derringer, G. (1994). A balancing act: optimizing a product's properties. *Quality Progress*.,**27**, 51-58.

Etemad, B. Whitcomb DC. (2001). Chronic pancreatitis: Diagnosis, classification, and new genetic developments. *Gastroenterology*, **120**(3), 682.

Harrington, E.C., Jr. (1965). The Desirability Function. *Industrial Quality Control* **21,** 494-498.

Hong, J., Suh, H.(2005). A Strategic Model for Consolidating BSC Measures Based on the Desirability Function: A Case Study of a Website Company
GESTS Int'l Trans. Computer Science and Engr., Vol.18, No.1 **99 (2005)**

Kalbfleisch, J. D. and Prentice, R. L. (2002). *The Statistical Analysis of Failure Time Data* New York: Wiley, 2nd Edition.

Kaplan, R.S., Norton, D.P. (2000). Using the Scorecards a Strategic Management System. The Havard Business Review

Kern, R.S., Nuechterlein, K H., Green, M.F., ET Al. (2008). The Matrics Consensus Cognitive Battery, Part 2: Co-norming and standardization. The American Journal of Psychiatry, 165(2), 214.

Kim, Kwang-Jae, and Lin, D.K.J., (2000). Simultaneous optimization of mechanical properties of steel by maximizing exponential desirability functions. *Appl. Statistics*,**49** (3), 311-325.

Khuri, A. I., and Conlon, M. (1981). Simultaneous Optimization of Multiple Responses Represented by Polynomial Regression Functions. *Technometrics*, **23**, 363-375.

Lankisch PG, Banks PA. Pancreatitis. Heidelberg, Germany: Springer-Verlag, 1998.

National Institute of Health. (1996).*Guidelines for the Care and Use of Laboratory Animals.* National Academy Press, Washington, D.C.

Nelder, J.A., Mead, R. (1965). A simplex method for function optimization. *The Computer Journal* **7**, 308-313.

Nichols, J. (2007). Defensive Composite Score. Retrieved March 2, 2009. Available from **http://www.82games.com/nichols1.htm**

Moser, V.C. (1991). Applications of a Neurobehavioral Screening Battery. *Journal of the American College of Toxicology* ,**10**, 661-669.

Moser, V.C., Tilson, H.A., MacPhail, R.C., Becking, G.C., Cuomo, V., Frantik, E., Kulig, B.M., and Winnwkw, G. (1997). The IPCS Collaborative Study on Neurobehavioral Screening Methods: II. Protocol Designs and Testing Procedures. *NeuroTocicology*, **18**,929-938.

Moser, V C. (2000). The functional observational battery in adult and developing rats. *Neurotoxicology*, **21**(6), 989.

Parker, S. M. and Gennings, C. (2008). Penalized locally optimal experimental designs for nonlinear models. *Journal of Agriculture, Biological and Environmental Statistics*, **13**(3):334-354.

Pizarro, C, & PIZARRO. (2006). Multiple response optimisation based on desirability functions of a microwave-assisted extraction method for the simultaneous determination of chloroanisoles and chlorophenols in oak barrel sawdust. *Journal of chromatography A*, **1132**(1-2), 8.

Rotz, L D. (2002). Public health assessment of potential biological terrorism agents. *Emerging Infectious Diseases*, **8**(2), 225.

Sandhu, B S, Hackworth, W A, Stevens, S, Bouhaidar, D S, Zfass, A M. (2007). Recurrent flares of pancreatitis predict development of exocrine insufficiency in chronic pancreatitis. *Clinical gastroenterology and hepatology*, **5**(9), 1085.

Sarles, H. (1991). Definitions and classifications of pancreatitis. *Pancreas*, **6**(4), 470.

Sarner, M. Cotton, P.B. (1984). Classification of pancreatitis. Cambridge 1983. *Gut,* **25**, 756.

SAS Institute Inc. (2005). SAS OnlineDoc ® 9.1.3, Cary, NC: SAS Institute Inc

Shih, M., Gennings, C., Chinchilli, V.M., and Carter, W.H.,Jr. (2003). Titrating and evaluating multi-drug regimens within subjects. *Statistics in Medicine,* **22**(14), 2257-2279.

Trautmann, H. and Weihs, T. (2006). On the distribution of the desirability index using harrington's desirability function. Metrika, 63(2), 207.

Uomo, G. (2002). How far are we from the most accurate classification system for chronic pancreatitis?. *Journal of the Pancreas* **3**(3), 62.

Weihs, C. (2007). Quality assurance for statistical consulting. *AStA, **91**,* 429-440.

Xu, K, Lin, D.K.J., Tang, L., Xie, M. (2004). Multiresponse systems optimization using a goal attainment approach. *IIE Transactions*, **36**(5), 433.

Appendix A

Appendix A

```
***********************************************************************
* This program in PROC IML of SAS conducts the Nelder-Mead simplex    *
* program for function minimization.  The program is adapted from     *
* Olsson (1974), Journal of Quality Technology 6, 53-57.              *
*                                                                     *
* The user needs to provide the module FUNCTION which contains the    *
* code for calculating the function given the set of parameters.  For *
* this module PARMS is the column K-vector of parameters and FN_VALUE *
* is the function evaluated at PARMS.  Also, the user needs to        *
* provide the column K-vectors of starting values IN_PARMS and        *
* initial step values IN_STEPS when calling this module.              *
*                                                                     *
* There is no printed output that results from running this module.   *
* However, the column K-vector PARMS (the set of parameters which     *
* minimize the function), FN_VALUE (the function evaluated at PARMS), *
* and COUNT (the number of iterations) are available to the user.     *
*                                                                     *
* As a cautionary note, the user should not construct matrices in     *
* PROC IML with the naming convention _MATRIX_ because the modules    *
* use this for all temporary matrices.                                *
***********************************************************************;

START SIMPLEX;
_NITER_=9000;_EPS_=1.0E-8;_K_=NROW(IN_PARMS);_KK_=_K_+1;
_P_=J(_K_,_KK_,0);_Y_=J(1,_KK_,0);
COUNT=0;_DABIT_=2.04607E-20;_BIGNUM_=1.0E38;_KONVGE_=5;
_PBAR_=J(_K_,1,0);_PSTAR_=_PBAR_;_P2STAR_=_PBAR_;
_RCOEFF_=1.0;_ECOEFF_=1.5;_CCOEFF_=0.5;
```

```
**CONSTRUCT INITIAL SIMPLEX**;

_P_[,_KK_]=IN_PARMS;PARMS=IN_PARMS;RUN FUNCTION;_A_=FN_VALUE;
_Y_[_KK_]=_A_;COUNT=COUNT+1;

*print 'initial estimates' count parms _A_;

DO _I_=1 TO _K_;
 _P_[,_I_]=IN_PARMS;_P_[_I_,_I_]=_P_[_I_,_I_]+IN_STEPS[_I_];
 _TEMP_=_P_[,_I_];PARMS=_TEMP_;RUN FUNCTION;_A_=FN_VALUE;
 _Y_[,_I_]=_A_;COUNT=COUNT+1;
END;

**SIMPLEX IS NOW CONSTRUCTED**;

HILO:
_YLO_=MIN(_Y_);_YNEWLO_=MAX(_Y_);
DO _I_=1 TO _KK_;
  IF _Y_[,_I_]=_YLO_ THEN _ILO_=_I_;
  IF _Y_[,_I_]=_YNEWLO_ THEN _IHI_=_I_;
END;

**PERFORM CONVERGENCE CHECK ON FUNCTION**;
**THE RATIO OF THE LARGEST TO SMALLEST VERTEX FUNCTION TEST**;

_DCHK_=(_YNEWLO_+_DABIT_)/(_YLO_+_DABIT_)-1;
IF ABS(_DCHK_)<_EPS_ THEN GOTO BEST;
_KONVGE_=_KONVGE_-1;
IF _KONVGE_=0 THEN DO;_KONVGE_=5;
  DO _I_=1 TO _K_;
```

```
    _COORD1_=_P_[_I_,1];_COORD2_=_COORD1_;
        DO _J_=2 TO _KK_;
           IF _P_[_I_,_J_]<_COORD1_ THEN _COORD1_=_P_[_I_,_J_];
           IF _P_[_I_,_J_]>_COORD2_ THEN _COORD2_=_P_[_I_,_J_];
        END;
        _DCHK_=(_COORD2_+_DABIT_)/(_COORD1_+_DABIT_)-1;
        IF ABS(_DCHK_)<=_EPS_ THEN GO TO BEST;
  END;
END;
IF COUNT>_NITER_ THEN GOTO BEST;

**CALCULATE _PBAR_, THE CENTROID OF THE**;
**SIMPLEX VERTICES EXCEPTING THAT WITH _Y_ VALUE _YNEWLO_**;

DO _I_=1 TO _K_;_Z_=0;
  DO _J_=1 TO _KK_;_Z_=_Z_+_P_[_I_,_J_];END;
 _Z_=_Z_-_P_[_I_,_IHI_];_PBAR_[_I_]=_Z_/_K_;
END;
_PSTAR_=(1+_RCOEFF_)*_PBAR_-_RCOEFF_*_P_[,_IHI_];

**REFLECTION THROUGH THE CENTROID**;

PARMS=_PSTAR_;RUN FUNCTION;_YSTAR_=FN_VALUE;
COUNT=COUNT+1;
IF _YSTAR_ >=_YLO_ THEN GOTO NOEXT;
IF COUNT >=_NITER_ THEN GOTO RETAIN;

**SUCCESSFUL REFLECTION, SO EXTENSION**;

_P2STAR_=_ECOEFF_*_PSTAR_+(1-_ECOEFF_)*_PBAR_;
PARMS=_P2STAR_;RUN FUNCTION;_Y2STAR_=FN_VALUE;
```

```
COUNT=COUNT+1;

**RETAIN EXTENSION OR CONTRACTION**;

IF _Y2STAR_ >=_YSTAR_ THEN GOTO RETAIN;

EXTCON:
_P_[,_IHI_]=_P2STAR_;
_Y_[_IHI_]=_Y2STAR_;
GOTO HILO;

**NO EXTENSION**;

NOEXT:
_L_=0;
DO _I_=1 TO _KK_;
  IF _Y_[_I_]>_YSTAR_ THEN _L_=_L_+1;
END;
IF _L_>1 THEN GOTO RETAIN;

**CONTRACTION ON THE REFLECTION SIDE OF THE CENTRIOD**;

IF _L_=1 THEN DO;
  _P_[,_IHI_]=_PSTAR_;
  _Y_[_IHI_]=_YSTAR_;
END;

**CONTRACTION ON THE _Y_[_IHI_] SIDE OF THE CENTROID**;

IF COUNT>=_NITER_ THEN GOTO BEST;
_P2STAR_=_CCOEFF_*_P_[,_IHI_]+(-_CCOEFF_+1)*_PBAR_;
```

```
PARMS= _P2STAR_;RUN FUNCTION;_Y2STAR_=FN_VALUE;COUNT=COUNT+1;
IF _Y2STAR_<_Y_[_IHI_] THEN GOTO EXTCON;

**CONTRACT THE WHOLE SIMPLEX**;

DO _J_=1 TO _KK_;
  DO _I_=1 TO _K_;
     _P_[_I_,_J_]=0.5*(_P_[_I_,_J_]+_P_[_I_,_ILO_]);
  END;_XMIN_=_P_[,_J_];
  PARMS= _XMIN_;RUN FUNCTION;_A_=FN_VALUE;_Y_[,_J_]=_A_;
END;
COUNT=COUNT+_KK_;
IF COUNT<_NITER_ THEN GOTO HILO;ELSE GOTO BEST;

RETAIN:
_P_[,_IHI_]=_PSTAR_;_Y_[_IHI_]=_YSTAR_;GOTO HILO;

BEST:
DO _J_=1 TO _KK_;_XMIN_=_P_[,_J_];
  PARMS= _XMIN_;RUN FUNCTION;_A_=FN_VALUE;_Y_[_J_]=_A_;
END;
_YNEWLO_=_BIGNUM_;
DO _J_=1 TO _KK_;
  IF _Y_[_J_]<_YNEWLO_ THEN DO;
    _YNEWLO_=_Y_[_J_];_IBEST_=_J_;
  END;
END;
_Y_[_IBEST_]=_BIGNUM_;_YSEC_=_BIGNUM_;
DO _J_=1 TO _KK_;
  IF _Y_[_J_]<_YSEC_ THEN DO;
    _YSEC_=_Y_[_J_];_ISEC_=_J_;
  END;
END;
_XMIN_=_P_[,_IBEST_];_XSEC_=_P_[,_ISEC_];
PARMS=_XMIN_;FN_VALUE=_YNEWLO_;

FREE _NITER_ _EPS_ _K_ _KK_ _P_ _Y_ _DABIT_ _BIGNUM_ _KONVGE_;
FREE _PBAR_ _PSTAR_ _P2STAR_ _RCOEFF_ _ECOEFF_ _CCOEFF_;
FREE _A_ _I_ _TEMP_ _YLO_ _YNEWLO_ _ILO_ _IHI_;
FREE _DCHK_ _COORD1_ _COORD2_ _Z_ _YSTAR_ _L_ _J_;
FREE _XMIN_ _IBEST_ _YSEC_ _XSEC_;
FINISH;
```

Appendix B

Appendix B

Optimal Box- Cox Transoformations

```
data desire3;
*set sasuser.Bota3data;

set dat.Botdatab3;


     if day=1 and time_of_day='pm' then do time=1.5;end;
     if day=2 and time_of_day='am' then do time=2.0;end;
     if day=2 and time_of_day='pm' then do time=2.5;end;
     if day=3 and time_of_day='am' then do time=3.0;end;
     if day=3 and time_of_day='pm' then do time=3.5;end;
     if day=4 and time_of_day='am' then do time=4.0;end;
     if day=4 and time_of_day='pm' then do time=4.5;end;
     if day=5 and time_of_day='am' then do time=5.0;end;

     act=(act1+act2)/2;
     res=(res1+res2)/2;
     mus=(mus1+mus2)/2;
     pilo=(pilo1+pilo2)/2;

     if act=1 then do d2=1;end;
     if act=1.5 then do d2=.9;end;
     if act=2 then do d2=.8;end;
     if act=2.5 then do d2=.65;end;
     if act=3 then do d2=.5;end;


     if res=1 then do d3=1;end;
     if res=1.5 then do d3=.725;end;
     if res=2 then do d3=.45;end;
     if res=2.5 then do d3=.4;end;
     if res=3 then do d3=.35;end;
     if res=3.5 then do d3=.25;end;
     if res=4 then do d3=.15;end;


     if mus=1 then do d4=1;end;
     if mus=1.5 then do d4=.8;end;
     if mus=2 then do d4=.75;end;
     if mus=2.5 then do d4=.575;end;
     if mus=3 then do d4=.40;end;


     if pilo=1 then do d5=1;end;
     if pilo=1.5 then do d5=.9;end;
     if pilo=2 then do d5=.8;end;


     DA_unwt = (d2*d3*d4*d5)**(1/4);
     if da_unwt = . then do; d2=9; d3=9; d4=9; d5=9; end;


run;




data new3;
     set dat.deathb3;
     death_time=0;
     censor=100;

      if death_day=1 and Death_Time_of_Day='PM' then do death_time=1.5; end;
      if death_day=2 and Death_Time_of_Day='AM' then do death_time=2.0; end;
      if death_day=2 and Death_Time_of_Day='PM' then do death_time=2.5; end;
      if death_day=3 and Death_Time_of_Day='AM' then do death_time=3.0; end;
      if death_day=3 and Death_Time_of_Day='PM' then do death_time=3.5; end;
      if death_day=4 and Death_Time_of_Day='AM' then do death_time=4.0; end;
      if death_day=4 and Death_Time_of_Day='PM' then do death_time=4.5; end;


     if id="M025" then do death_time=3.5;end;
     if id="M025" then do death_day=3;end;
     if id="N021" then do death_time=1.5;end;
     if id="N022" then do death_time=1.5;end;
     if id="N025" then do death_time=1.5;end;
     if death_day=' ' then do   death_time=5.0; end;

     if death_time=5.0 then censor=0; else censor=1; *censored is 0;
 run;


data dinfo3;
 set new3;
 keep id dose death_time censor;
run;
proc sort data=desire3; by id;
proc sort data=dinfo3; by id;
data all3;
```

```
  merge desire3 dinfo3; by id;

run;
proc sort data=all3; by  death_time descending censor id;


data da1 da2 da3 da4 da5 da6 da7 da8;
   set all3;
   if time=1.5 then output da1;
   if time=2.0 then output da2;
   if time=2.5 then output da3;
   if time=3.0 then output da4;
   if time=3.5 then output da5;
   if time=4.0 then output da6;
   if time=4.5 then output da7;
   if time=5.0 then output da8;
run;

proc means data=da1 n;
by death_time descending censor;
output out=di  n=di;
run;

data da1;
   merge da1 di; by death_time descending censor;
   d2i=di*censor;
   if first.death_time=1 then first_death=1;else first_death=0;
run;
```

```
%macro nlp_dopt(step,w2,w3,w4);

* D-optimal Design;
proc iml;
title' ';
  use da1; read all var{d2 d3 d4 d5} into da1;
  use da2; read all var{d2 d3 d4 d5} into da2;
  use da3; read all var{d2 d3 d4 d5} into da3;
  use da4; read all var{d2 d3 d4 d5} into da4;
  use da5; read all var{d2 d3 d4 d5} into da5;
  use da6; read all var{d2 d3 d4 d5} into da6;
  use da7; read all var{d2 d3 d4 d5} into da7;
  use da8; read all var{d2 d3 d4 d5} into da8;
  use da1;
    read all var {censor}into cens;
     read all var {death_time} into time;
     read all var {d2i} into di;
     read all var {first_death}into first;


start initial;
      w2=&w2; w3=&w3; w4=&w4; w5=4-w2-w3-w4;
      wsum=w2+w3+w4+w5;
      w=(w2//w3//w4//w5);
      censfirst=cens#first;
      N=nrow(da1);

finish initial;
run initial;
```

```
/***************************************************************
 defining the log likelihood function

****************************************************************/
Start ll(betaest) global(bigx,cens,beta, first,da1,da2,da3,da4,da5,da6,da7,
da8,di,time,covb,w,wsum,m2lik);

betaest=betaest`;
N=nrow(da1);
lik=0;
deriv=j(nrow(betaest),nrow(betaest),0);
derivnew=j(nrow(betaest),nrow(betaest),0);
censfirst=cens#first;

do i=1 to N;

     if censfirst[i]=1 then do;


          if i=1 then rs=J(N-i+1,1,1);
               else rs=J(i-1,1,0)//J(N-i+1,1,1);

if time[i]= 1.5 then _DA_= (exp(log(DA1)*w))##(1/wsum);
if time[i]= 2   then _DA_= (exp(log(DA2)*w))##(1/wsum);
if time[i]= 2.5 then _DA_= (exp(log(DA3)*w))##(1/wsum);
if time[i]= 3   then _DA_= (exp(log(DA4)*w))##(1/wsum);
if time[i]= 3.5 then _DA_= (exp(log(DA5)*w))##(1/wsum);
if time[i]= 4   then _DA_= (exp(log(DA6)*w))##(1/wsum);
if time[i]= 4.5 then _DA_= (exp(log(DA7)*w))##(1/wsum);
if time[i]= 5   then _DA_= (exp(log(DA8)*w))##(1/wsum);
```

```
  _DA_=1-_DA_; * implement 1-DA;
  bigx = _da_||(_da_)##2;


  term=bigx*betaest;

  eterm=exp(term#rs);

  p1=((di[i]#eterm# rs#bigx)`* (rs#bigx))/(rs`*eterm);
  p2= ((rs#bigx)`* eterm)/(rs`*eterm);
  dip2= ((di[i]#rs#bigx)`* eterm)/(rs`*eterm);
  dnew=(p1-(dip2*p2`));
  derivnew=derivnew+dnew;
  dsn=design(time);
  s_matrix=bigx`*dsn;
  snew=dsn*s_matrix`;

  loglik=snew[i,]*betaest-(di[i]#log(eterm`*rs));

  lik=lik+loglik;

  newlik=lik;

     end;
end;



m2lik=-2*(newlik);
```

```
return(newlik);
finish ll;





/***********************************
Starting Values
* starting values should be a row vector;
**********************************/

beta0={20, -23};

/************************************
  Options
************************************/
optn=j(1,11,.);
optn[1]=1; *min=0 max=1;
optn[2]=0; *Controls the amount of printout;
/************************************
Termination Criteria
************************************/
tc=j(1,13,.);
tc[1]=5000; *maximum iterations;


/***********************************
     Control parameter vector
***********************************/
```

```
par=j(1,10,.);
par[2]=1E-1; * initial step length;
par[6]=0.05;*required accuracy of the line search;
/**********************************
Call procedure
**********************************/
*run ll;
start varcov;


call nlpnrr(rc, betaest, 'll', beta0, optn);
call nlpfdd(f,g,h,"ll",betaest);

var=inv(-h);
se=sqrt(vecdiag(var));
*print betaest var se;
gvar=det(var);
dopt=gvar;* needs to be dopt=gvar;
finish varcov;


run varcov;


%include simplex;

start function;
     w2=parms[1];w3=parms[2];w4=parms[3];
     w5=4-w2-w3-w4;

     w=(w2//w3//w4//w5);
     *print parms;
     *contraints;
     if((w2<0)+(w3<0)+(w4<0)+(w5<0) +(w2>4)+(w3>4)
          +(w4>4) )>0 then fn_value=10**30;
               else do;
               run varcov;
               fn_value=dopt;

end;
finish;

Start Optima;
     in_parms=(w2//w3//w4);
     in_steps=in_parms*&step;
     run varcov;
     se_beta = sqrt(vecdiag(var));
     print "intial evaluation" w2 w3 w4 w5 , "Var-cov" var, 'dopt' dopt;
     print "Initial Beta Est:" betaest se_beta m2lik;
     run simplex;
     run function;
     se_beta = sqrt(vecdiag(var));
     print "The Final Weights :" w2 w3 w4 w5 fn_value,
            'With' count  ,'Variance' var;
            print 'Beta Est:' betaest se_beta m2lik;
finish;

run optima;
quit;

%mend;
* nlp_dopt(step,w2,w3,w4);
```

Appendix C

Appendix C

```
* Gompertz Optimization

/***********************************************************************************
***
 Defining the log likelihood function
************************************************************************************
**/
Start ll(betaest) global(bigx,cens,beta, first, b02,b12,b03,b13,b04,
b14,b05,b15,a1,r1,
m1,p1,a2,r2,m2,p2,a3,r3,m3,p3,a4,r4,m4,p4,a5,r5,m5,p5,a6,r6,m6,p6, a7,r7,m7,p7,
a8,r8, m8,p8,di,time,covb,w,wsum,m2lik,DAL);

      betaest=betaest`;
      N=nrow(a1);
      lik=0;
      deriv=j(nrow(betaest),nrow(betaest),0);
      derivnew=j(nrow(betaest),nrow(betaest),0);
      censfirst=cens#first;

      do i=1 to N;

          if censfirst[i]=1 then do;

            if i=1 then rs=J(N-i+1,1,1);
               else rs=J(i-1,1,0)//J(N-i+1,1,1);

                    if time[i]= 1.5 then do;
                         d2l=exp(-exp(-(b02+b12*a1)));
                         d3l=exp(-exp(-(b03+b13*r1)));
                         d4l=exp(-exp(-(b04+b14*m1)));
```

```
                  d5l=exp(-exp(-(b05+b15*p1)));

end;
             if time[i]= 2  then do;
                  d2l=exp(-exp(-(b02+b12*a2)));
                  d3l=exp(-exp(-(b03+b13*r2)));
                  d4l=exp(-exp(-(b04+b14*m2)));
                  d5l=exp(-exp(-(b05+b15*p2)));

             end;

             if time[i]= 2.5 then do;
                  d2l=exp(-exp(-(b02+b12*a3)));
                  d3l=exp(-exp(-(b03+b13*r3)));
                  d4l=exp(-exp(-(b04+b14*m3)));
                  d5l=exp(-exp(-(b05+b15*p3)));

             end;

             if time[i]= 3 then do;
                  d2l=exp(-exp(-(b02+b12*a4)));
                  d3l=exp(-exp(-(b03+b13*r4)));
                  d4l=exp(-exp(-(b04+b14*m4)));
                  d5l=exp(-exp(-(b05+b15*p4)));

             end;

             if time[i]= 3.5 then do;
                  d2l=exp(-exp(-(b02+b12*a5)));
                  d3l=exp(-exp(-(b03+b13*r5)));
                  d4l=exp(-exp(-(b04+b14*m5)));
```

```
     d5l=exp(-exp(-(b05+b15*p5)));

end;

if time[i]= 4 then do;
     d2l=exp(-exp(-(b02+b12*a6)));
     d3l=exp(-exp(-(b03+b13*r6)));
     d4l=exp(-exp(-(b04+b14*m6)));
     d5l=exp(-exp(-(b05+b15*p6)));

end;

if time[i]= 4.5 then do;
     d2l=exp(-exp(-(b02+b12*a7)));
     d3l=exp(-exp(-(b03+b13*r7)));
     d4l=exp(-exp(-(b04+b14*m7)));
     d5l=exp(-exp(-(b05+b15*p7)));

end;

if time[i]= 5 then do;
     d2l=exp(-exp(-(b02+b12*a8)));
     d3l=exp(-exp(-(b03+b13*r8)));
     d4l=exp(-exp(-(b04+b14*m8)));
     d5l=exp(-exp(-(b05+b15*p8)));


end;

     DAL=(d2l#d3l#d4l#d5l)##(1/4);
     DAold=DAL;
```

```
                         DAL=1-DAL;

                         bigx =DAL||(DAL)##2;

                         term=bigx*betaest;
                         eterm=exp(term#rs);

                         pp1=((di[i]#eterm# rs#bigx)`* (rs#bigx))/(rs`*eterm);
                         pp2= ((rs#bigx)`* eterm)/(rs`*eterm);
                         dip2= ((di[i]#rs#bigx)`* eterm)/(rs`*eterm);
                         dnew=(pp1-(dip2*pp2`));
                         derivnew=derivnew+dnew;
                         dsn=design(time);
                         s_matrix=bigx`*dsn;
                         snew=dsn*s_matrix`;

                         loglik=snew[i,]*betaest-(di[i]#log(eterm`*rs));
                         lik=lik+loglik;
                         newlik=lik;

       end;
end;



     m2lik=-2*(newlik); * -2loglikelihood;
     return(newlik);
finish ll;
*run ll;
```

```
/***********************************************************************************
**********************************
Starting Values

* starting values should be a row vector;
************************************************************************************
*********************************/

beta0={3,-7};

/***********************************************************************************
*********************************
  Options
************************************************************************************
*********************************/
optn=j(1,11,.);
optn[1]=1; *min=0 max=1;
optn[2]=0; *Controls the amount of printout;

/***********************************************************************************
**********************************
Termination Criteria
************************************************************************************
**********************************/
tc=j(1,13,.);
tc[1]=5000; *maximum iterations;
```

```
/***********************************************************************************
**********************************
     Control parameter vector
************************************************************************************
**********************************/

par=j(1,10,.);
par[2]=1E-1; * initial step length;
par[6]=0.05;*required accuracy of the line search;

/***********************************************************************************
**********************************
Call procedure
************************************************************************************
**********************************/

start varcov;

     betaest=beta0`;
     call nlpnrr(rc, betaest, 'll', beta0, optn);
     call nlpfdd(f,g,h,"ll",betaest);

     var=ginv(-h);
     se=sqrt(vecdiag(var));

     theta1=betaest[1,1];
     theta2=betaest[1,2];

     se1=se[1,1];
     se2=se[2,1];
     tm1=(theta1/se1);
     tm2=(theta2/se2);

     sumsq=(tm1**2)+(tm2**2);


     dopt=-sumsq;* negative to maximize the objective function;


finish varcov;


run varcov;



%include simplex;

start function;
     b02=parms[1]; b12=parms[2]; b03=parms[3]; b13=parms[4]; b04=parms[5];
     b14=parms[6]; b05=parms[7];b15=parms[8];

     if ((b02>10)+(b03>10)+(b04>10)+(b05>10)+(b12>0)+(b13>0)+(b14>0)+(b15>0))>0 then
     fn_value=1000000000000000000000000000;

          else do;
               run varcov;

                temp=DAL[loc(DAL<1)]; * creates a subsetted vector temp that
                containes    values of DAL <1.;
```

```
                 diff=temp[<>]-temp[><];* trying to spread the DAL scale out;

                if (diff<0.3)>0 then  fn_value=1000000000000000000000000000;
                     else
                          fn_value=dopt; *print count betaest fn_value;
     end;

finish;

Start Optima;
     in_parms=(b02//b12//b03//b13//b04//b14//b05//b15);
     parms=in_parms;
     in_steps=in_parms*.2;
          run varcov;

     print "intial evaluation" b02 b12 b03 b13 b04 b14 b05 b15  , "Var-cov" var,
     'dopt'
            dopt;
     print "Initial Beta Est:" betaest m2lik ;*se_beta;
          run simplex;
          run function;

     print "The Final Weights :" b02 b12 b03 b13 b04 b14 b05 b15  fn_value,
           'With' count  ,'Variance' var;
     print 'Beta Est:' betaest m2lik;
     create DAL from DAL; append from DAL;

finish;
run optima;
quit;
```

Appendix D

Appendix D

```
* Logistic Optimization;

/***************************************************************
 defining the log likelihood function

****************************************************************/
Start ll(betaest) global(bigx,cens,beta,
first,b02,b12,b03,b13,b04,b14,b05,b15,a1,r1,m1,p1,a2,r2,m2,p2,a3,r3,m3,p3,

a4,r4,m4,p4,a5,r5,m5,p5,a6,r6,m6,p6,a7,r7,m7,p7,a8,r8,m8,p8,di,time,covb,w,wsum,m2li
k);

       betaest=betaest`;
       N=nrow(a1);
      lik=0;
      deriv=j(nrow(betaest),nrow(betaest),0);
      derivnew=j(nrow(betaest),nrow(betaest),0);
      censfirst=cens#first;

      do i=1 to N;

         if censfirst[i]=1 then do;


            if i=1 then rs=J(N-i+1,1,1);
                else rs=J(i-1,1,0)//J(N-i+1,1,1);

                              * this calulates the overall Desriability Function for
each time point;
```

```
     if time[i]= 1.5 then do;
          d2l=(1+exp(-(b02+b12*a1)))##-1;
          d3l=(1+exp(-(b03+b13*r1)))##-1;
          d4l=(1+exp(-(b04+b14*m1)))##-1;
          d5l=(1+exp(-(b05+b15*p1)))##-1;

     end;
if time[i]= 2   then do;
          d2l=(1+exp(-(b02+b12*a2)))##-1;
          d3l=(1+exp(-(b03+b13*r2)))##-1;
          d4l=(1+exp(-(b04+b14*m2)))##-1;
          d5l=(1+exp(-(b05+b15*p2)))##-1;

     end;

    if time[i]= 2.5 then do;
          d2l=(1+exp(-(b02+b12*a3)))##-1;
          d3l=(1+exp(-(b03+b13*r3)))##-1;
          d4l=(1+exp(-(b04+b14*m3)))##-1;
          d5l=(1+exp(-(b05+b15*p3)))##-1;

     end;
    if time[i]= 3   then do;
          d2l=(1+exp(-(b02+b12*a4)))##-1;
          d3l=(1+exp(-(b03+b13*r4)))##-1;
          d4l=(1+exp(-(b04+b14*m4)))##-1;
          d5l=(1+exp(-(b05+b15*p4)))##-1;

     end;

    if time[i]= 3.5 then do;
```

```
      d2l=(1+exp(-(b02+b12*a5)))##-1;
      d3l=(1+exp(-(b03+b13*r5)))##-1;
      d4l=(1+exp(-(b04+b14*m5)))##-1;
      d5l=(1+exp(-(b05+b15*p5)))##-1;

 end;

if time[i]= 4   then do;
      d2l=(1+exp(-(b02+b12*a6)))##-1;
      d3l=(1+exp(-(b03+b13*r6)))##-1;
      d4l=(1+exp(-(b04+b14*m6)))##-1;
      d5l=(1+exp(-(b05+b15*p6)))##-1;


 end;

if time[i]= 4.5 then do;
      d2l=(1+exp(-(b02+b12*a7)))##-1;
      d3l=(1+exp(-(b03+b13*r7)))##-1;
      d4l=(1+exp(-(b04+b14*m7)))##-1;
      d5l=(1+exp(-(b05+b15*p7)))##-1;

 end;

if time[i]= 5 then do;
      d2l=(1+exp(-(b02+b12*a8)))##-1;
      d3l=(1+exp(-(b03+b13*r8)))##-1;
      d4l=(1+exp(-(b04+b14*m8)))##-1;
      d5l=(1+exp(-(b05+b15*p8)))##-1;
```

```
                              end;
                                   DAl=(d2l#d3l#d4l#d5l)##(1/4);
                                   bigx =DAl||(DAl)##2;

                    term=bigx*betaest;

                    eterm=exp(term#rs);

                    pp1=((di[i]#eterm# rs#bigx)`* (rs#bigx))/(rs`*eterm);
                    pp2= ((rs#bigx)`* eterm)/(rs`*eterm);
                    dip2= ((di[i]#rs#bigx)`* eterm)/(rs`*eterm);
                    dnew=(pp1-(dip2*pp2`));
                    derivnew=derivnew+dnew;
                    dsn=design(time);
                    s_matrix=bigx`*dsn;
                    snew=dsn*s_matrix`;

                    loglik=snew[i,]*betaest-(di[i]#log(eterm`*rs));

                   lik=lik+loglik;

                    newlik=lik;

     end;
end;



m2lik=-2*(newlik);
return(newlik);
finish ll;
```

```
/**********************************
Starting Values
* starting values should be a row vector;
**********************************/
*beta0={2.2115216, -21.54134};

beta0={20,-23};
/***********************************
  Options
***********************************/
optn=j(1,11,.);
optn[1]=1; *min=0 max=1;
optn[2]=0; *Controls the amount of printout;

/***********************************
Termination Criteria
***********************************/
tc=j(1,13,.);
tc[1]=5000; *maximum iterations;


/**********************************
     Control parameter vector
***********************************/
par=j(1,10,.);
par[2]=1E-1; * initial step length;
par[6]=0.05;*required accuracy of the line search;

/**********************************
```

```
Call procedure
**********************************/
*run ll;
start varcov;


     call nlpnrr(rc, betaest, 'll', beta0, optn);
     call nlpfdd(f,g,h,"ll",betaest);

     var=ginv(-h);

     gvar=det(var);
     dopt=gvar;

finish varcov;


run varcov;



%include simplex;

start function;
     b02=parms[1]; b12=parms[2]; b03=parms[3]; b13=parms[4]; b04=parms[5];
b14=parms[6]; b05=parms[7];b15=parms[8];


*contraints;
if ((b02>10)+(b03>10)+(b04>10)+(b05>10)+(b12>0)+(b13>0)+(b14>0)+(b15>0))>0 then
fn_value=1000000000000000000000000000;
```

```
     else do;
          run varcov;
          fn_value=dopt; *print count betaest fn_value;
end;

finish;

Start Optima;
     in_parms=(b02//b12//b03//b13//b04//b14//b05//b15);
     in_steps=in_parms*.4; * may need to change step sizes;
          run varcov;

     print "intial evaluation" b02 b12 b03 b13 b04 b14 b05 b15 , "Var-cov" var,
'dopt' dopt;
     print "Initial Beta Est:" betaest m2lik;*se_beta;
          run simplex;
          run function;

     print "The Final Weights :" b02 b12 b03 b13 b04 b14 b05 b15  fn_value, 'With'
count  ,'Variance' var;
     print 'Beta Est:' betaest m2lik;

finish;

run optima;
quit;
```

Appendix E

Appendix E

```
--------------------------------------------------------------------------------------
Chapter 1 Optimum Scale Code
***********************************************************************************;

data desire3;
*set sasuser.Bota3data;

set dat.Botdatab3;


     if day=1 and time_of_day='pm' then do time=1.5;end;
     if day=2 and time_of_day='am' then do time=2.0;end;
     if day=2 and time_of_day='pm' then do time=2.5;end;
     if day=3 and time_of_day='am' then do time=3.0;end;
     if day=3 and time_of_day='pm' then do time=3.5;end;
     if day=4 and time_of_day='am' then do time=4.0;end;
     if day=4 and time_of_day='pm' then do time=4.5;end;
     if day=5 and time_of_day='am' then do time=5.0;end;
     *if day=5 and time_of_day='pm' then do time=5.5;*end;

     act=(act1+act2)/2;
     res=(res1+res2)/2;
     mus=(mus1+mus2)/2;
     pilo=(pilo1+pilo2)/2;

     theta1=0.8;
     theta2=0.5;
     theta3=0.45;
     theta4=0.35;
     theta5=0.15;
```

```
theta7=0.75;
theta8=0.40;
theta9=0.8;


if act=1 then do d2=1;end;
if act=1.5 then do d2=(1+theta1)/2;end;
if act=2 then do d2=theta1;end;
if act=2.5 then do d2=(theta1+theta2)/2;end;
if act=3 then do d2=theta2;end;


if res=1 then do d3=1;end;
if res=1.5 then do d3=(1+theta3)/2;end;
if res=2 then do d3=theta3;end;
if res=2.5 then do d3=(theta3+theta4)/2;end;
if res=3 then do d3=theta4;end;
if res=3.5 then do d3=(theta4+theta5)/2;end;
if res=4 then do d3=theta5;end;


if mus=1 then do d4=1;end;
if mus=1.5 then do d4=(1+theta7)/2;end;
if mus=2 then do d4=theta7;end;
if mus=2.5 then do d4=(theta7+theta8)/2;end;
if mus=3 then do d4=theta8;end;


if pilo=1 then do d5=1;end;
if pilo=1.5 then do d5=(1+theta9)/2;end;
if pilo=2 then do d5=theta9;end;
```

```
     DA_unwt = (d2*d3*d4*d5)**(1/4);
     if da_unwt = . then do; d2=9; d3=9; d4=9; d5=9; end;


run;

data new3;
     set dat.deathb3;
     death_time=0;
     censor=100;
     if death_day=1 and Death_Time_of_Day='PM' then do death_time=1.5; end;
     if death_day=2 and Death_Time_of_Day='AM' then do death_time=2.0; end;
     if death_day=2 and Death_Time_of_Day='PM' then do death_time=2.5; end;
     if death_day=3 and Death_Time_of_Day='AM' then do death_time=3.0; end;
     if death_day=3 and Death_Time_of_Day='PM' then do death_time=3.5; end;
     if death_day=4 and Death_Time_of_Day='AM' then do death_time=4.0; end;
     if death_day=4 and Death_Time_of_Day='PM' then do death_time=4.5; end;


     if id="M025" then do death_time=3.5;end;
     if id="M025" then do death_day=3;end;
     if id="N021" then do death_time=1.5;end;
     if id="N022" then do death_time=1.5;end;
     if id="N025" then do death_time=1.5;end;
     if death_day=' ' then do   death_time=5.0; end;

     if death_time=5.0 then censor=0; else censor=1; *censored is 0;
 run;
```

```
data dinfo3;
      set new3;
      keep id dose death_time censor;
run;
proc sort data=desire3; by id;
proc sort data=dinfo3; by id;
data all3;
  merge desire3 dinfo3; by id;
run;
proc sort data=all3; by  death_time descending censor id;


data da1 da2 da3 da4 da5 da6 da7 da8;
   set all3;
   if time=1.5 then output da1;
   if time=2.0 then output da2;
   if time=2.5 then output da3;
   if time=3.0 then output da4;
   if time=3.5 then output da5;
   if time=4.0 then output da6;
   if time=4.5 then output da7;
   if time=5.0 then output da8;
run;

proc means data=da1 n noprint;
     by death_time descending censor;
     output out=di  n=di;
run;
```

```
data da1;
   merge da1 di; by death_time descending censor;
   d2i=di*censor;
   if first.death_time=1 then first_death=1;else first_death=0;
run;


* D-optimal Design;
* Macro for optimum scale parameters.  Input step size and scale parameters to be
optimized;

%macro dopt_scale(step,theta1,theta2,theta3,theta4,theta5,theta7,theta8,theta9);
proc iml;
title' ';

  use da1; read all var{act res mus pilo} into da1;
             act1=da1[,1]; res1=da1[,2]; mus1=da1[,3]; pilo1=da1[,4];
  use da2; read all var{act res mus pilo} into da2;
             act2=da2[,1]; res2=da2[,2]; mus2=da2[,3]; pilo2=da2[,4];
  use da3; read all var{act res mus pilo} into da3;
             act3=da3[,1]; res3=da3[,2]; mus3=da3[,3]; pilo3=da3[,4];
  use da4; read all var{act res mus pilo} into da4;
             act4=da4[,1]; res4=da4[,2]; mus4=da4[,3]; pilo4=da4[,4];
  use da5; read all var{act res mus pilo} into da5;
             act5=da5[,1]; res5=da5[,2]; mus5=da5[,3]; pilo5=da5[,4];
  use da6; read all var{act res mus pilo} into da6;
             act6=da6[,1]; res6=da6[,2]; mus6=da6[,3]; pilo6=da6[,4];
  use da7; read all var{act res mus pilo} into da7;
             act7=da7[,1]; res7=da7[,2]; mus7=da7[,3]; pilo7=da7[,4];
  use da8; read all var{act res mus pilo} into da8;
```

```
             act8=da8[,1]; res8=da8[,2]; mus8=da8[,3]; pilo8=da8[,4];


  use da1;
     read all var {censor}into cens;
     read all var {death_time} into time;
     read all var {d2i} into di;
     read all var {first_death}into first;





start initial;

     theta1=&theta1;
     theta2=&theta2;
     theta3=&theta3;
     theta4=&theta4;
     theta5=&theta5;
     theta7=&theta7;
     theta8=&theta8;
     theta9=&theta9;

     censfirst=cens#first;
     N=nrow(da1);

finish initial;
run initial;
```

```
/***************************************************************
 Defining the log likelihood function


****************************************************************/

Start ll(betaest) global(bigx,cens,beta,
     first,da1,theta1,theta2,theta3,theta4,theta5,theta7,theta8,theta9,
     da1_1,da1_2,da1_3,da1_4,da2_1,da2_2,da2_3,da2_4,da3_1,da3_2,da3_3,da3_4,da4_1,d
     a4_2,da4_3,da4_4,da5_1,da5_2,da5_3,da5_4,
     da6_1,da6_2,da6_3,da6_4,da7_1,da7_2,da7_3,da7_4,da8_1,da8_2,da8_3,da8_4,
     act1,act2,act3,act4,act5,act6,act7act8,res1,res2,res3,res4,res5,res6,res7,res8,
     mus1,mus2,mus3,mus4,mus5,mus6,mus7,mus8,pilo1,pilo2,pilo3,pilo4,pilo5,pilo6,pil
     o7,pilo8,di,time,covb,w,wsum,m2lik);

     betaest=betaest`;
     N=nrow(da1);
     nsum=ncol(da1);
     lik=0;
     deriv=j(nrow(betaest),nrow(betaest),0);
     derivnew=j(nrow(betaest),nrow(betaest),0);
     censfirst=cens#first;
               *print 'before do loop' censfirst di time;
          do i=1 to N;

               if censfirst[i]=1 then do;


                    if i=1 then rs=J(N-i+1,1,1);
                         else rs=J(i-1,1,0)//J(N-i+1,1,1);
```

```
                        *print'just b4 update' i;

                              if time[i]= 1.5 then do;

                    ************************************
                                        Time 1.5;

                                   * activity at time 1.5;
                                   da1_1=((act1=1)#1)+
                                   ((act1=1.5)#((1+theta1)/2))+
                                   ((act1=2)#theta1)+
                                   ((act1=2.5)#((theta1+theta2)/2))+
                                   ((act1=3)#theta2);

                                   *respiration at time 1.5;

da1_2=(res1=1)+((res1=1.5)#((1+theta3)/2))+((res1=2)#theta3)+((res1=2.5)#((thet
a3+theta4)/2))+((res1=3)#theta4)+((res1=3.5)#((theta4+theta5)/2))+((res1=4)#theta5);

                                   *muscle tone at time 1.5;
da1_3=(mus1=1)+((mus1=1.5)#((1+theta7)/2))+((mus1=2)#theta7)+((mus1=2.5)#
((theta7+theta8)/2))+((mus1=3)#theta8);

                                   *piloerection at time 1.5;

da1_4=(pilo1=1)+((pilo1=1.5)#((1+theta9)/2))+((pilo1=2)#theta9);

                    _DA_= (da1_1#da1_2#da1_3#da1_4)##(1/nsum);

         end;
```

```
          if time[i]= 2   then do;

                              ************************************
                                 Time 2;

                              * activity at time 2;
            da2_1=((act2=1)#1)+((act2=1.5)#((1+theta1)/2))+((act2=2)#theta1)+
           ((act2=2.5)#((theta1+theta2)/2))+((act2=3)#theta2);

                              *respiration at time 2;

da2_2=(res2=1)+((res2=1.5)#((1+theta3)/2))+((res2=2)#theta3)+((res2=2.5)#
((theta3+theta4)/2))+((res2=3)#theta4)+((res2=3.5)#((theta4+theta5)/2))+
((res2=4)#theta5);

                              *muscle tone at time 2;
da2_3=(mus2=1)+((mus2=1.5)#((1+theta7)/2)) + ((mus2=2)#theta7)+

((mus2=2.5)#((theta7+theta8)/2))+((mus2=3)#theta8);

                              *piloerection at time 2;

da2_4=(pilo2=1)+((pilo2=1.5)#((1+theta9)/2))+((pilo2=2)#theta9);

               _DA_= (da2_1#da2_2#da2_3#da2_4)##(1/nsum);

          end;
          if time[i]= 2.5 then do;
                         ************************************
                                 Time 2.5;

                              * activity at time 2.5;
da3_1=((act3=1)#1)+((act3=1.5)#((1+theta1)/2))+((act3=2)#theta1)+
((act3=2.5)#((theta1+theta2)/2))+((act3=3)#theta2);

                              *respiration at time 2.5;

da3_2=(res3=1)+((res3=1.5)#((1+theta3)/2))+((res3=2)#theta3)+((res3=2.5)#((thet
a3+theta4)/2))+((res3=3)#theta4)+((res3=3.5)#((theta4+theta5)/2))+((res3=4)#
theta5);

                              *muscle tone at time 2.5;
da3_3=(mus3=1)+((mus3=1.5)#((1+theta7)/2))+((mus3=2)#theta7)+((mus3=2.5)#((thet
a7+theta8)/2))+((mus3=3)#theta8);

                              *piloerection at time 2.5;

da3_4=(pilo3=1)+((pilo3=1.5)#((1+theta9)/2))+((pilo3=2)#theta9);

                    _DA_= (da3_1#da3_2#da3_3#da3_4)##(1/nsum);
          end;

          if time[i]= 3   then do;
                              ************************************
                                 Time 3;
                              * activity at time 3;
da4_1=((act4=1)#1)+((act4=1.5)#((1+theta1)/2))+((act4=2)#theta1)+
((act4=2.5)#((theta1+theta2)/2))+((act4=3)#theta2);

                              *respiration at time 3;

da4_2=(res4=1)+((res4=1.5)#((1+theta3)/2))+((res4=2)#theta3)+((res4=2.5)#((thet
a3+theta4)/2))+((res4=3)#theta4)+((res4=3.5)#((theta4+theta5)/2))+((res4=4)
#theta5);

                              *muscle tone at time 3;
da4_3=(mus4=1)+ ((mus4=1.5)#((1+theta7)/2))+((mus4=2)#theta7)+
((mus4=2.5)#((theta7+theta8)/2))+((mus4=3)#theta8);

                              *piloerection at time 3;

da4_4=(pilo4=1)+((pilo4=1.5)#((1+theta9)/2))+((pilo4=2)#theta9);

               _DA_= (da4_1#da4_2#da4_3#da4_4)##(1/nsum);
          end;


          if time[i]= 3.5 then do;

                                   ************************************
                                      Time 3.5;

                                   * activity at time 3.5;
da5_1=((act5=1)#1)+((act5=1.5)#((1+theta1)/2))+((act5=2)#theta1)+
((act5=2.5)#((theta1+theta2)/2))+((act5=3)#theta2);
```

```
                                   *respiration at time 3.5;

da5_2=(res5=1)+((res5=1.5)#((1+theta3)/2))+((res5=2)#theta3)+((res5=2.5)#((thet
a3+theta4)/2))+((res5=3)#theta4)+((res5=3.5)#((theta4+theta5)/2))+((res5=4)#the
ta5);

                                   *muscle tone at time 3.5;
da5_3=(mus5=1)+((mus5=1.5)#((1+theta7)/2))+((mus5=2)#theta7)+
((mus5=2.5)#((theta7+theta8)/2))+((mus5=3)#theta8);

                                   *piloerection at time 3.5;

da5_4=(pilo5=1)+((pilo5=1.5)#((1+theta9)/2))+((pilo5=2)#theta9);

               _DA_= (da5_1#da5_2#da5_3#da5_4)##(1/nsum);
          end;


          if time[i]= 4   then do;

                                   ************************************
                                      Time 4;

                                   * activity at time 4;
da6_1=((act6=1)#1)+((act6=1.5)#((1+theta1)/2))+((act6=2)#theta1)+
((act6=2.5)#((theta1+theta2)/2))+((act6=3)#theta2);

                                   *respiration at time 4;

da6_2=(res6=1)+((res6=1.5)#((1+theta3)/2))+((res6=2)#theta3)+((res6=2.5)#((thet
a3+theta4)/2))+((res6=3)#theta4)+((res6=3.5)#((theta4+theta5)/2))+((res6=4)#the
ta5);

                                   *muscle tone at time 4;
da6_3=(mus6=1)+((mus6=1.5)#((1+theta7)/2)) + ((mus6=2)#theta7)+
((mus6=2.5)#((theta7+theta8)/2))+((mus6=3)#theta8);

                                   *piloerection at time 4;

da6_4=(pilo6=1)+((pilo6=1.5)#((1+theta9)/2))+((pilo6=2)#theta9);

               _DA_= (da6_1#da6_2#da6_3#da6_4)##(1/nsum);
          end;


          if time[i]= 4.5 then do;

                                   ************************************
                                      Time 4.5;

                                   * activity at time 4.5;
da7_1=((act7=1)#1)+(act7=1.5)#((1+theta1)/2))+((act7=2)#theta1)+((act7=2.5)#
((theta1+theta2)/2))+((act7=3)#theta2);

                                   *respiration at time 4.5;

da7_2=(res7=1)+((res7=1.5)#((1+theta3)/2))+((res7=2)#theta3)+((res7=2.5)#((thet
a3+theta4)/2))+((res7=3)#theta4)+((res7=3.5)#((theta4+theta5)/2))+((res7=4)#the
ta5);
```

```
                                   *muscle tone at time 4.5;
da7_3=(mus7=1)+ ((mus7=1.5)#((1+theta7)/2)) + ((mus7=2)#theta7)+
((mus7=2.5)#((theta7+theta8)/2))+((mus7=3)#theta8);

                                   *piloerection at time 4.5;

da7_4=(pilo7=1)+((pilo7=1.5)#((1+theta9)/2))+((pilo7=2)#theta9);

                  _DA_= (da7_1#da7_2#da7_3#da7_4)##(1/nsum);
          end;

          if time[i]= 5   then do;


************************************
                                      Time 5;

                                   * activity at time 5;
da8_1=((act8=1)#1)+((act8=1.5)#((1+theta1)/2))+((act8=2)#theta1)+
((act8=2.5)#((theta1+theta2)/2))+((act8=3)#theta2);

                                   *respiration at time 5;

da8_2=(res8=1)+((res8=1.5)#((1+theta3)/2))+((res8=2)#theta3)+((res8=2.5)#
((theta3+theta4)/2))+((res8=3)#theta4)+((res8=3.5)#((theta4+theta5)/2))+
((res8=4)#
theta5);

                                   *muscle tone at time 5;
```

```
     da8_3=(mus8=1)+ ((mus8=1.5)#((1+theta7)/2)) + ((mus8=2)#theta7)+

     ((mus8=2.5)#((theta7+theta8)/2))+((mus8=3)#theta8);

                                        *piloerection at time 5;

     da8_4=(pilo8=1)+((pilo8=1.5)#((1+theta9)/2))+((pilo8=2)#theta9);

                    _DA_= (da8_1#da8_2#da8_3#da8_4)##(1/nsum);
               end;

                    _DA_=1-_DA_; * implement 1-DA;
                    bigx = _da_||(_da_)##2;
                    term=bigx*betaest;
                    eterm=exp(term#rs);
                    p1=((di[i]#eterm# rs#bigx)`* (rs#bigx))/(rs`*eterm);
                    p2= ((rs#bigx)`* eterm)/(rs`*eterm);
                    dip2= ((di[i]#rs#bigx)`* eterm)/(rs`*eterm);
                    dnew=(p1-(dip2*p2`));
                    derivnew=derivnew+dnew;
                    dsn=design(time);
                    s_matrix=bigx`*dsn;
                    snew=dsn*s_matrix`;

                    loglik=snew[i,]*betaest-(di[i]#log(eterm`*rs));

                    lik=lik+loglik;
                    newlik=lik;

     end;
end;
```

```
m2lik=-2*(newlik);
return(newlik);
finish ll;

/**********************************
Starting Values
* starting values should be a row vector;
**********************************/
beta0={5, -10};
/***********************************
  Options
************************************/
optn=j(1,11,.);
optn[1]=1; *min=0 max=1;
optn[2]=0; *Controls the amount of printout;
/***********************************
Termination Criteria
************************************/
tc=j(1,13,.);
tc[1]=5000; *maximum iterations;


/**********************************
     Control parameter vector
***********************************/
par=j(1,10,.);
par[2]=1E-1; * initial step length;
par[6]=0.05;*required accuracy of the line search;
/**********************************
```

```
Call procedure
**********************************/
*run ll;
start varcov;
call nlpnrr(rc, betaest, 'll', beta0, optn);
call nlpfdd(f,g,h,"ll",betaest);

var=inv(-h);
se=sqrt(vecdiag(var));
gvar=det(var);
dopt=gvar;
finish varcov;


run varcov;




%include simplex;

start function;
theta1=parms[1];theta2=parms[2];theta3=parms[3];theta4=parms[4];theta5=parms[5];
theta7=parms[6];theta8=parms[7];theta9=parms[8];

*contraints;
if((theta1<=0)+(theta2<=0)+(theta3<=0)+(theta4<=0)+(theta5<=0)+(theta7<=0)+(theta8<=
0)+(theta9<=0)+(theta1>1)+(theta2>1)+(theta3>1)+(theta4>1)+(theta5>1)+(theta7>1)+
(theta8>1)+(theta9>1) )>0 then fn_value=10**30;
 else do;
     run varcov;
     fn_value=dopt;
```

```
end;
finish;

Start Optima;
     in_parms=(theta1//theta2//theta3//theta4//theta5//theta7//theta8//theta9);
     in_steps=in_parms*&step;
     run varcov;
     se_beta = sqrt(vecdiag(var));
     print "intial evaluation" theta1 theta2 theta3 theta4 theta5 theta7 theta8
     theta9 , "Var-cov" var, 'dopt' dopt;
     print "Initial Beta Est:" betaest se_beta m2lik;
     run simplex;
     run function;
     se_beta = sqrt(vecdiag(var));
     print "The Final Weights :" theta1 theta2 theta3 theta4 theta5  theta7 theta8
     theta9 fn_value,
            'With' count  ,'Variance' var;
            print 'Beta Est:' betaest se_beta m2lik;
finish;

run optima;
quit;

%mend;
```

Appendix F

Appendix F

Chapter 3 Penalized Optimality Code

```
/**********************************
Starting Values
* starting values should be a row vector;
**********************************/
beta0={20, -23};
/***********************************
  Options
************************************/
optn=j(1,11,.);
optn[1]=1; *min=0 max=1;
optn[2]=0; *Controls the amount of printout;
/***********************************
Termination Criteria
************************************/
tc=j(1,13,.);
tc[1]=5000; *maximum iterations;


/**********************************
     Control parameter vector
***********************************/
par=j(1,10,.);
par[2]=1E-1; * initial step length;
par[6]=0.05;*required accuracy of the line search;
/**********************************
Call procedure
**********************************/
```

```
start varcov;

call nlpnrr(rc, betaest, 'll', beta0, optn);
call nlpfdd(f,g,h,"ll",betaest);

var=inv(-h);
se=sqrt(vecdiag(var));
gvar=det(var);
dopt=gvar;* needs to be dopt=gvar;
finish varcov;


run varcov;




%include simplex;

start function;
w2=parms[1];w3=parms[2];w4=parms[3];
w5=4-w2-w3-w4;

w=(w2//w3//w4//w5);

     ylowmax=0.1; yhimax=0.2; gmax=0.1;
     amax=(ylowmax+yhimax)/2;
     bmax=(yhimax-ylowmax)/(2*log((1-gmax)/gmax));


     ylowmin=5.5; yhimin=7.5; gmin=0.1;
     amin=(ylowmin+yhimin)/2;
     bmin=(yhimin-ylowmin)/(2*log((1-gmin)/gmin));

     minw=min(w2,w3,w4,w5);

     dmax=(1+exp(-((minw-amax)/bmax)))**-1;
     dt=dmax;




if((w2<0)+(w3<0)+(w4<0)+( w5<0)+((w2+w3+w4+w5)>4) )>0 then fn_value=10**30;
else do;


run varcov;

       scalefactor=81.90;
       constant=lambda*scalefactor;
       desterm = constant*(1-dt);
       fn_value = dopt + desterm;

end;
finish;



Start Optima;
in_parms=(w2//w3//w4//w5);
in_steps=in_parms*&step;
run varcov;
se_beta = sqrt(vecdiag(var));
```

```
run simplex;
run function;
se_beta = sqrt(vecdiag(var));
finish;



start grid;
design=0;


do lambda=0 to 1 by 0.5;

do initw2=.5 to 1 by .5;
  initw3=.5;
  initw4=.5;
  initw5=4-initw2-initw3-initw4;


          w2=initw2; w3=initw3; w4=initw4;w5=initw5;
          w=(w2||w3||w4||w5)`;
          run optima;
          design=design+1;
          labels={'design'  'initw2' 'initw3' 'initw4' 'initw5'
                    'dopt' 'lambda' 'constant' 'count' 'w2' 'w3' 'w4' 'w5' 'dt'
                    'desterm' 'fn_value' };
          results=results//(design||in_parms[1]||in_parms[2]||in_parms[3]
          ||in_parms[4]||dopt||lambda||constant||count||w2||w3||w4||w5||dt||desterm|
          |fn_value);
     end;
     end;
```

```
          create results from results[colname=labels]; append from results;
     finish;
     run grid;

quit;

proc sort data=results; by lambda fn_value;run;
proc print data=results;by lambda;run;

data final;
     set results; by lambda;
     if first.lambda;
proc print;
     goptions ftext=script htext=1.8;
     symbol1 i=join l=1 c=blue;
     symbol2 i=join l=3 c=red;

     axis1 label=(a=90 font=simulate height=1.5 'D (solid line, blue)')
     value=(font=simulate height=1.3);
     axis2 label=(a=270 font=simulate height=1.5 'Generalized Variance (dashed
     line,red)' ) value=(font=simulate height=1.3);
     proc gplot;
     plot dt*lambda/vaxis=axis1 ;
     plot2 dopt*lambda/vaxis=axis2;
     label lambda='l';
     title ' ';
run; quit;

%mend;
```

Appendix G

Appendix G

*Chater 4 Optimal Transformation Code;

```
data dat.bimal12_15p1;
     set sasuser.bimal1;

     informat Date_of_Ist_MRI MONYY5. Date_of_Ist_MRI MONYY5.
     Date_of_Exocrine_failure_yrs_ MONYY5.;
     format Date_of_Ist_MRI MONYY5. Date_of_Ist_MRI MONYY5.
     Date_of_Exocrine_failure_yrs_ MONYY5.;
run;


data dat.bimal12_15p2;
     set sasuser.bimal1_2;
     informat Date_of_Ist_MRI MONYY5. Date_of_Ist_MRI MONYY5.
     Date_of_Exocrine_failure_yrs_ MONYY5.;
     format Date_of_Ist_MRI MONYY5. Date_of_Ist_MRI MONYY5.
     Date_of_Exocrine_failure_yrs_ MONYY5.;
run;

data bimal12_15p1;
     set dat.bimal12_15p1;


     if SBE_3_1='x' then do SBE_3_1=1;end;
     if SBE___3_1='x' then do SBE___3_1=1;end;
     if SBE_3_1=' ' and SBE___3_1=' ' then do SBE_3_1='.';SBE___3_1='.';end; * or
=0;

     if SBE_3_1=' ' then do SBE_3_1=0;end;
     if SBE___3_1=' ' then do SBE___3_1=0;end;


     if SBE_3_1='n ' then do SBE_3_1='.';end;
     if SBE___3_1='n ' then do SBE___3_1='.';end;


     SBE_L31=SBE_3_1+0;
     SBE_G31=SBE___3_1+0;


     if SBE___3_2='x' then do SBE___3_2=1;end;
     if SBE___3_20='x' then do SBE___3_20=1;end;

     if SBE___3_2='y' then do SBE___3_2=1;end;
     if SBE___3_20='y' then do SBE___3_20=1;end;

     if SBE___3_2=' ' and SBE___3_20=' ' then do SBE___3_2='.';SBE___3_20='.';end;

     if SBE___3_2=' ' then do SBE___3_2=0;end;
     if SBE___3_20=' ' then do SBE___3_20=0;end;


     if SBE___3_2='n ' then do SBE___3_2='.';end;
     if SBE___3_20='n ' then do SBE___3_20='.';end;
     SBE_L32=SBE___3_2+0;
     SBE_G32=SBE___3_20+0;


     if MPD_irregularity_1='Y' or MPD_irregularity_1='y'  then do
     D_irregularity_1=1;end;
     else if MPD_irregularity_1="N" or MPD_irregularity_1="n"
          then do MPD_irregularity_1=0;end;
               else if MPD_irregularity_1=" "
                    then do MPD_irregularity_1=".";end;
                         MPD_i1=MPD_irregularity_1+0;* converting character to
numeric;

     if MPD_irregularity_2='Y' or MPD_irregularity_2='y'
          then do MPD_irregularity_2=1;end;
               else if MPD_irregularity_2="N" or MPD_irregularity_2="n"
                    then do MPD_irregularity_2=0;end;
                         else if MPD_irregularity_2=" "
                              then do MPD_irregularity_2=".";end;
                                   *converting character to numeric;
                                   MPD_i2=MPD_irregularity_2+0;


     if MPD_stricture_1='Y' or MPD_stricture_1='y'
          then do MPD_stricture_1=1;end;
               else if MPD_stricture_1="N" or MPD_stricture_1="n"
                     then do MPD_stricture_1=0;end;
                         else if MPD_stricture_1=" "
                              then do MPD_stricture_1=".";end;
                                   *converting character to numeric;
                              MPD_s1=MPD_stricture_1+0;

if MPD_stricture_2='Y' or MPD_stricture_2='y'
     then do MPD_stricture_2=1;end;
          else if MPD_stricture_2="N" or MPD_stricture_2="n"
               then do MPD_stricture_2=0;end;
                    else if MPD_stricture_2=" "
                         then do MPD_stricture_2=".";end;
                              *converting character to numeric;
                              MPD_s2=MPD_stricture_2+0;


if MPD_calculi_1='Y' or MPD_calculi_1='y'
     then do MPD_calculi_1=1;end;
          else if MPD_calculi_1="N" or MPD_calculi_1="n"
               then do MPD_calculi_1=0;end;
                    else if MPD_calculi_1=" "
                         then do MPD_calculi_1=".";end;
                              MPD_c1=MPD_calculi_1+0;

if MPD_calculi_2='Y' or MPD_calculi_2='y'
     then do MPD_calculi_2=1;end;
          else if MPD_calculi_2="N" or MPD_calculi_2="n"
               then do MPD_calculi_2=0;end;
                    else if MPD_calculi_2=" "
                         then do MPD_calculi_2=".";end;
                              MPD_c2=MPD_calculi_2+0;



if Pseudocyst_1='Y' or Pseudocyst_1='y'
```

```
     then do Pseudocyst_1=1;end;
          else if Pseudocyst_1="N" or Pseudocyst_1="n"
               then do Pseudocyst_1=0;end;
                    else if Pseudocyst_1=" "
                         then do Pseudocyst_1=".";end;
                              Psc1=Pseudocyst_1+0;


if Pseudocyst_2='Y' or Pseudocyst_2='y'
     then do Pseudocyst_2=1;end;
          else if Pseudocyst_2="N" or Pseudocyst_2="n"
               then do Pseudocyst_2=0;end;
                    else if Pseudocyst_2=" "
                         then do Pseudocyst_2=".";end;
                              Psc2=Pseudocyst_2+0;

if MPD_leak_1='Y' or MPD_leak_1='y'
     then do MPD_leak_1=1;end;
          else if MPD_leak_1="N" or MPD_leak_1="n"
               then do MPD_leak_1=0;end;
                    else if MPD_leak_1=" "
                         then do MPD_leak_1=".";end;
                              MPD_L1=MPD_leak_1+0;


if MPD_leak_2='Y' or MPD_leak_2='y'
     then do MPD_leak_2=1;end;
          else if MPD_leak_2="N" or MPD_leak_2="n"
               then do MPD_leak_2=0;end;
                    else if MPD_leak_2=" "
                         then do MPD_leak_2=".";end;
```

```
                              MPD_L2=MPD_leak_2+0;


if Pancreatic_atrophy_1='Y' or Pancreatic_atrophy_1='y'
     then do Pancreatic_atrophy_1=1;end;
          else if Pancreatic_atrophy_1="N" or Pancreatic_atrophy_1="n"
               then do Pancreatic_atrophy_1=0;end;
                    else if Pancreatic_atrophy_1=" "
                         then do Pancreatic_atrophy_1=".";end;
                              PA1=Pancreatic_atrophy_1+0;

if Pancreatic_atrophy_2='Y' or Pancreatic_atrophy_2='y'
     then do Pancreatic_atrophy_2=1;end;
          else if Pancreatic_atrophy_2="N" or Pancreatic_atrophy_2="n"
               then do Pancreatic_atrophy_2=0;end;
                    else if Pancreatic_atrophy_2=" "
                         then do Pancreatic_atrophy_2=".";end;
                              PA2=Pancreatic_atrophy_2+0;

if Contour_abnormality_of_bile_duct='Y' or Contour_abnormality_of_bile_duct='y'
     then do Contour_abnormality_of_bile_duct=1;end;
          else if Contour_abnormality_of_bile_duct="N" or
          Contour_abnormality_of_bile_duct="n"
               then do Contour_abnormality_of_bile_duct=0;end;
                    else if Contour_abnormality_of_bile_duct=" "
                         then do Contour_abnormality_of_bile_duct=".";end;
                              CABD1=Contour_abnormality_of_bile_duct+0;

if Contour_abnormality_of_bile_duc0='Y' or Contour_abnormality_of_bile_duc0='y'
     then do Contour_abnormality_of_bile_duc0=1;end;
```

```
               else if Contour_abnormality_of_bile_duc0="N" or
               Contour_abnormality_of_bile_duc0="n"
                    then do Contour_abnormality_of_bile_duc0=0;end;
                              else if Contour_abnormality_of_bile_duc0=" "
                                   then do
                              Contour_abnormality_of_bile_duc0=".";end;

CABD2=Contour_abnormality_of_bile_duc0+0;


if Bile_duct_stricture_1='Y' or Bile_duct_stricture_1='y'
     then do Bile_duct_stricture_1=1;end;
          else if Bile_duct_stricture_1="N" or Bile_duct_stricture_1="n"
               then do Bile_duct_stricture_1=0;end;
                    else if Bile_duct_stricture_1=" "
                         then do Bile_duct_stricture_1=".";end;
                              BDS1=Bile_duct_stricture_1+0;

if Bile_duct_stricture_2='Y' or Bile_duct_stricture_2='y'
     then do Bile_duct_stricture_2=1;end;
          else if Bile_duct_stricture_2="N" or Bile_duct_stricture_2="n"
               then do Bile_duct_stricture_2=0;end;
                    else if Bile_duct_stricture_2=" "
                         then do Bile_duct_stricture_2=".";end;

BDS2=Bile_duct_stricture_2+0;

SBS1=Side_branch_size__mm__1;
SBS2=Side_branch_size__mm__2;

MPD_size1=MPD_size__mm__1;
```

```
     MPD_size2=MPD_size__mm__2;

     cstage1=CAMBRIDGE_STAGE_1;
     cstage2=CAMBRIDGE_STAGE_2;






     drop SBE_3_1 SBE___3_1 SBE___3_2 SBE___3_20 MPD_irregularity_1
     MPD_irregularity_2
          MPD_calculi_1 MPD_calculi_2 Pseudocyst_1 Pseudocyst_2 MPD_leak_1
          MPD_leak_2 Pancreatic_atrophy_1 Pancreatic_atrophy_2
          Contour_abnormality_of_bile_duct Contour_abnormality_of_bile_duc0
          Bile_duct_stricture_1 Bile_duct_stricture_2 Side_branch_size__mm__1
          Side_branch_size__mm__2 MPD_size__mm__1 MPD_size__mm__2 CAMBRIDGE_STAGE_1
          CAMBRIDGE_STAGE_2 MPD_stricture_1 MPD_stricture_2
          Duration_of_symptoms_days_ Duration_of_symptoms
          Duration_of_symptoms_years_
          Time_to_DM_years_  Time_to_Exo_failure;
run;


data bimal12_15p2;
     set dat.bimal12_15p2;


     SBE_L31=SBE_3_1+0;
     SBE_G31=SBE___3_1+0;
```

```
SBE_L32=SBE___3_2+0;
SBE_G32=SBE___3_20+0;


MPD_i1=MPD_irregularity_1+0;* converting character to numeric;

MPD_i2=MPD_irregularity_2+0;* converting character to numeric;

MPD_s1=MPD_stricture_1+0;* converting character to numeric;


MPD_s2=MPD_stricture_2+0;* converting character to numeric;

MPD_c1=MPD_calculi_1+0;

MPD_c2=MPD_calculi_2+0;

Psc1=Pseudocyst_1+0;

Psc2=Pseudocyst_2+0;

MPD_L1=MPD_leak_1+0;

MPD_L2=MPD_leak_2+0;


PA1=Pancreatic_atrophy_1+0;
```

```
     PA2=Pancreatic_atrophy_2+0;

     CABD1=Contour_abnormality_of_bile_duct+0;

     CABD2=Contour_abnormality_of_bile_duc0+0;


     BDS1=Bile_duct_stricture_1+0;

     BDS2=Bile_duct_stricture_2+0;

     SBS1=Side_branch_size__mm__1;
     SBS2=Side_branch_size__mm__2;

     MPD_size1=MPD_size__mm__1;
     MPD_size2=MPD_size__mm__2;

     cstage1=CAMBRIDGE_STAGE_1;
     cstage2=CAMBRIDGE_STAGE_2;



     drop SBE_3_1 SBE___3_1 SBE___3_2 SBE___3_20 MPD_irregularity_1
MPD_irregularity_2
          MPD_calculi_1 MPD_calculi_2 Pseudocyst_1 Pseudocyst_2 MPD_leak_1
          MPD_leak_2 Pancreatic_atrophy_1 Pancreatic_atrophy_2
          Contour_abnormality_of_bile_duct Contour_abnormality_of_bile_duc0
          Bile_duct_stricture_1 Bile_duct_stricture_2 Side_branch_size__mm__1
          Side_branch_size__mm__2 MPD_size__mm__1 MPD_size__mm__2 CAMBRIDGE_STAGE_1
          CAMBRIDGE_STAGE_2 MPD_stricture_1 MPD_stricture_2
          Duration_of_symptoms_days_ Duration_of_symptoms
          Duration_of_symptoms_years_
          Time_to_DM_years_  Time_to_Exo_failure;

     run;


data bimal12_15;
     set bimal12_15p1 bimal12_15p2;
     MRI_date=Date_of_Ist_MRI;
     DM_date=Date_of_onset_of_DM;
     EXO_date=Date_of_Exocrine_failure_yrs_;

     bsline_DM=DM_date-MRI_date;
     bsline_EXO=EXO_date-MRI_date;

     format MRI_date MONYY5. DM_date MONYY5. EXO_date MONYY5.;

     if bsline_DM>0 or bsline_DM='.' then do bsline_DM=0;end;
     else do bsline_DM=1;end;

     if bsline_EXO>0 or bsline_EXO='.' then do bsline_EXO=0;end;
     else do bsline_EXO=1;end;

     lookdate='01DEC08'd;
     format lookdate MONYY5.;

     DM_now=lookdate-DM_date;
     if DM_now='.' then do DMnow=0;end; else do DMnow=1;end;

     EXO_now=lookdate-EXO_date;
     if EXO_now='.' then do EXOnow=0;end; else do EXOnow=1;end;

      Psc_now=0;
      BDS_now=0;

     if Psc1=1 or Psc2=1 then do Psc_now=1;end;

     if BDS1=1 or BDS2=1 then do BDS_now=1;end;


     bsl_scr=bsline_EXO+bsline_DM+Psc1+ BDS1;

     flu_scr=DMnow+ EXOnow+Psc_now+BDS_now;
     any1=flu_scr>1;

run;


data fluscore;
     set bimal12_15;

     sbs=SBS1;
     ylowmin=2; yhimin=3;     g1min=0.15;
     a1min=(yhimin+ylowmin)/2;
     b1min=(yhimin-ylowmin)/(2*log((1-g1min)/g1min));
     d12s=(1+exp((sbs-a1min)/b1min))**-1; * Smaller -the-better;



     logit_95=log((.95)/(1-.95));
     logit_08=log((.8)/(1-.8));
```

```
m=-(logit_95-logit_08)/3;
dt1=1/(1+exp(-(4+m*MPD_size1)));


ylowmin1=2; yhimin1=2.5;      g1min1=0.2;
a1min1=(yhimin1+ylowmin1)/2;
b1min1=(yhimin1-ylowmin1)/(2*log((1-g1min1)/g1min1));
dt2=(1+exp(-(MPD_size1-a1min1)/b1min1))**-1;* bigger-the-better;

d13=dt1*dt2;


If PA1=1 then do d5=1;end;
If PA1=0 then do d5=0;end;


if MPD_s1=1 then do d9=0.5;end;
if MPD_s1=0 then do d9=1;end;

if MPD_l1=1 then do d8=0.5;end;
if MPD_l1=0 then do d8=1;end;

if concomittant_alcohol_use =1 then do d1=0.5;end;
if concomittant_alcohol_use =0 then do d1=1;end;

if ongoing_smoking=1 then do d2=1;end;
if ongoing_smoking=0 then do d2=0.5;end;

if SBE_G31=1 then do d4=0.5;end;
if SBE_G31=0 then do d4=1;end;
```

```
     if MPD_i1=1 then do d3=0.5;end;
     if MPD_i1=0 then do d3=1;end;

     if MPD_c1=1 then do d6=0.5;end;
     if MPD_c1=0 then do d6=1;end;

     if CABD1=1 then do d11=0.5;end;
     if CABD1=0 then do d11=1;end;


     ds13=(d12s*d8*d9*d1*d11*d4*d3*d6*d13)**(1/9);
     dnew=ds13*10;
     dnew=ds13;

run;

/***********************************************************************************
****************/


proc iml;
     use fluscore;
     read all var{dnew} into x;
     read all var{flu_scr} into R;
     read all var{d12s} into d1;
     read all var{d8} into d2;
     read all var{d9} into d3;
     read all var{d1} into d4;
     read all var{d11} into d5;
     read all var{d4} into d6;
read all var{d3} into d7;
read all var{d6} into d8;
read all var{d13} into d9;



start initial;

     alpha={2.2397 0.9029 -1.2476 -3.4281}; beta=-0.1647;

     a1=alpha[,1];
     a2=alpha[,2];
     a3=alpha[,3];
     a4=alpha[,4];


     w1=1; w2=1; w3=1; w4=1; w5=1; w6=1; w7=1; w8=1; w9=1;
     wsum=w1+w2+w3+w4+w5+w6+w7+w8+w9;
     w=(w1//w2//w3//w4//w5//w6//w7//w8//w9);

     N=nrow(x);

finish initial;
run initial;


start ll(abeta)
     global(d1,d2,d3,d4,d5,d6,d7,d8,d9,w1,w2,w3,w4,w5,w6,w7,w8,w9,
     wsum,p1,p2,p3,p4,R);
```

```
          S=((d1##w1)#(d2##w2)#(d3##w3)#(d4##w4)#(d5##w5)#(d6##w6)#(d7##w7)#(d8##d8)
          #(
          d9##w9))##(1/wsum);
          S_scale=S;

          *Pi’s for the likelihood equations;
          p4=(exp(abeta[4]+abeta[5]*S_scale))/(1+(exp(abeta[4]+abeta[5]*S_scale)));
          p3=(exp(abeta[3]+abeta[5]*S_scale))/(1+(exp(abeta[3]+abeta[5]*S_scale)));
          p2=(exp(abeta[2]+abeta[5]*S_scale))/(1+(exp(abeta[2]+abeta[5]*S_scale)));
          p1=(exp(abeta[1]+abeta[5]*S_scale))/(1+(exp(abeta[1]+abeta[5]*S_scale)));
          pi4=p4;*P(Y>4);
          pi3=p3-p4;*P(Y>3)-P(Y>4);
          pi2=p2-p3;*P(Y>2)-P(Y>3);
          pi1=p1-p2;*P(Y>1)-P(Y>2);
          pi0=1-p1;


     sum=(R=0)#log(pi0)+(R=1)#log(pi1)+(R=2)#log(pi2)+(R=3)#log(pi3)+(R=4)#log(pi4);
          sum_sum=sum(sum);
          m2lik=-2*sum_sum;*-2LL;
          return(sum_sum);
     finish ll;

/**********************************************************************************
*****
Starting Values
* starting values should be a row vector;
***********************************************************************************
****/
```

```
beta0={2.2397, 0.9029, -1.2476, -3.4281,-0.1647};

/***********************************************************************************
***
  Options
************************************************************************************
***/
optn=j(1,11,.);
optn[1]=1; *min=0 max=1;
optn[2]=0; *Controls the amount of printout;

/***********************************************************************************
***
Termination Criteria
************************************************************************************
***/
tc=j(1,13,.);
tc[1]=5000; *maximum iterations;


/***********************************************************************************
**
     Control parameter vector
************************************************************************************
**/
par=j(1,10,.);
par[2]=1E-1; * initial step length;
par[6]=0.05;*required accuracy of the line search;
```

```
/**********************************
Call procedure
**********************************/

start varcov;

     call nlpnrr(rc, abeta, 'll', beta0, optn);
     call nlpfdd(f,g,h,"ll",abeta);


     var=inv(-h);
     se=sqrt(vecdiag(var));
     gvar=det(var);
     dopt=gvar;
finish varcov;

run varcov;



%include simplex;

start function;
w1=parms[1];w2=parms[2];w3=parms[3];w4=parms[4];w5=parms[5];w6=parms[6];w7=parms[7];
w8=parms[8]; w9=9-w1-w2-w3-w4-w5-w6-w7-w8;
w=(w1//w2//w3//w4//w5//w6//w7//w8);

if((w1<0)+(w2<0)+(w3<0)+(w4<0)+( w5<0)+( w6<0)+( w7<0)+( w8<0)+( w9<0)
   +((w1+w2+w3+w4+w5+w6+w7+w8+w9)>9) )>0 then fn_value=10**30;
else do;
run varcov;
```

```
fn_value=dopt; *print count betaest fn_value;

end;
finish;

Start Optima;
in_parms=(w1//w2//w3//w4//w5//w6//w7//w8//w9);
in_steps=in_parms*.5;
run varcov;
se_betas = sqrt(vecdiag(var));
run simplex;
run function;


se_betas = sqrt(vecdiag(var));
finish;


start grid;
     design=0;
     min_fn=10**10;



 do initw1=0.5 to 2 by .5;
 do initw2=0.5 to 2 by .5;
     initw3=1;* to 2 by .5;
     initw4=1;* to 2 by .5;
     initw5=1;*.5 to 1 by .5;
     initw6=1;*.5 to 1 by .5;
     initw7=1;*.5 to 1 by .5;
     initw8=1;*.5 to 1 by .5;

     initw9=9-initw1-initw2-initw3-initw4-initw5-initw6-initw7-initw8;

          w1=initw1; w2=initw2; w3=initw3; w4=initw4;
          w5=initw5;w6=initw6;w7=initw7;w8=initw8; w9=initw9;
          w=(w1||w2||w3||w4||w5||w6||w7||w8||w9)`;

          run optima;

          design=design+1;
               if fn_value<min_fn then do;
                    min_fn=fn_value;
                    mincasecount=count;
                    minin_parms=in_parms;
                    wmin=w1||w2||w3||w4||w5
                    ||w6||w7||w8||w9;
               end;
          end;
          end;

finish;

          run grid;
          print "Fn_value" min_fn  "Count" mincasecount;
          print "Initial Weights" minin_parms;
          print "Final Weights" wmin;


quit;
```

Appendix H

Appendix H

Chapter 4 Penalized Optimality Code

```
data dat.bimal12_15p1;
     set sasuser.bimal1;

     informat Date_of_Ist_MRI MONYY5. Date_of_Ist_MRI MONYY5.
     Date_of_Exocrine_failure_yrs_ MONYY5.;
     format Date_of_Ist_MRI MONYY5. Date_of_Ist_MRI MONYY5.
     Date_of_Exocrine_failure_yrs_ MONYY5.;
run;


data dat.bimal12_15p2;
     set sasuser.bimal1_2;
     informat Date_of_Ist_MRI MONYY5. Date_of_Ist_MRI MONYY5.
     Date_of_Exocrine_failure_yrs_ MONYY5.;
     format Date_of_Ist_MRI MONYY5. Date_of_Ist_MRI MONYY5.
     Date_of_Exocrine_failure_yrs_ MONYY5.;
run;

data bimal12_15p1;
     set dat.bimal12_15p1;


     if SBE_3_1='x' then do SBE_3_1=1;end;
     if SBE___3_1='x' then do SBE___3_1=1;end;

     if SBE_3_1=' ' and SBE___3_1=' ' then do SBE_3_1='.';SBE___3_1='.';end; * or
=0;
```

```
if SBE_3_1=' ' then do SBE_3_1=0;end;
if SBE___3_1=' ' then do SBE___3_1=0;end;


if SBE_3_1='n ' then do SBE_3_1='.';end;
if SBE___3_1='n ' then do SBE___3_1='.';end;


SBE_L31=SBE_3_1+0;
SBE_G31=SBE___3_1+0;


if SBE___3_2='x' then do SBE___3_2=1;end;
if SBE___3_20='x' then do SBE___3_20=1;end;

if SBE___3_2='y' then do SBE___3_2=1;end;
if SBE___3_20='y' then do SBE___3_20=1;end;

if SBE___3_2=' ' and SBE___3_20=' ' then do SBE___3_2='.';SBE___3_20='.';end;

if SBE___3_2=' ' then do SBE___3_2=0;end;
if SBE___3_20=' ' then do SBE___3_20=0;end;


if SBE___3_2='n ' then do SBE___3_2='.';end;
if SBE___3_20='n ' then do SBE___3_20='.';end;



SBE_L32=SBE___3_2+0;
```

```
     SBE_G32=SBE___3_20+0;


     if MPD_irregularity_1='Y' or MPD_irregularity_1='y'  then do
     D_irregularity_1=1;end;
     else if MPD_irregularity_1="N" or MPD_irregularity_1="n"
          then do MPD_irregularity_1=0;end;
               else if MPD_irregularity_1=" "
                    then do MPD_irregularity_1=".";end;
                         MPD_i1=MPD_irregularity_1+0;* converting character to
numeric;

     if MPD_irregularity_2='Y' or MPD_irregularity_2='y'
          then do MPD_irregularity_2=1;end;
               else if MPD_irregularity_2="N" or MPD_irregularity_2="n"
                    then do MPD_irregularity_2=0;end;
                         else if MPD_irregularity_2=" "
                              then do MPD_irregularity_2=".";end;
                                   *converting character to numeric;
                                   MPD_i2=MPD_irregularity_2+0;


     if MPD_stricture_1='Y' or MPD_stricture_1='y'
          then do MPD_stricture_1=1;end;
               else if MPD_stricture_1="N" or MPD_stricture_1="n"
                     then do MPD_stricture_1=0;end;
                         else if MPD_stricture_1=" "
                              then do MPD_stricture_1=".";end;
                                   *converting character to numeric;
                                   MPD_s1=MPD_stricture_1+0;
```

```
if MPD_stricture_2='Y' or MPD_stricture_2='y'
     then do MPD_stricture_2=1;end;
          else if MPD_stricture_2="N" or MPD_stricture_2="n"
               then do MPD_stricture_2=0;end;
                    else if MPD_stricture_2=" "
                         then do MPD_stricture_2=".";end;
                              *converting character to numeric;
                              MPD_s2=MPD_stricture_2+0;


if MPD_calculi_1='Y' or MPD_calculi_1='y'
     then do MPD_calculi_1=1;end;
          else if MPD_calculi_1="N" or MPD_calculi_1="n"
               then do MPD_calculi_1=0;end;
                    else if MPD_calculi_1=" "
                         then do MPD_calculi_1=".";end;
                              MPD_c1=MPD_calculi_1+0;

if MPD_calculi_2='Y' or MPD_calculi_2='y'
     then do MPD_calculi_2=1;end;
          else if MPD_calculi_2="N" or MPD_calculi_2="n"
               then do MPD_calculi_2=0;end;
                    else if MPD_calculi_2=" "
                         then do MPD_calculi_2=".";end;
                              MPD_c2=MPD_calculi_2+0;



if Pseudocyst_1='Y' or Pseudocyst_1='y'
     then do Pseudocyst_1=1;end;
          else if Pseudocyst_1="N" or Pseudocyst_1="n"
```

```
               then do Pseudocyst_1=0;end;
                    else if Pseudocyst_1=" "
                         then do Pseudocyst_1=".";end;
                              Psc1=Pseudocyst_1+0;


if Pseudocyst_2='Y' or Pseudocyst_2='y'
     then do Pseudocyst_2=1;end;
          else if Pseudocyst_2="N" or Pseudocyst_2="n"
               then do Pseudocyst_2=0;end;
                    else if Pseudocyst_2=" "
                         then do Pseudocyst_2=".";end;
                              Psc2=Pseudocyst_2+0;

if MPD_leak_1='Y' or MPD_leak_1='y'
     then do MPD_leak_1=1;end;
          else if MPD_leak_1="N" or MPD_leak_1="n"
               then do MPD_leak_1=0;end;
                    else if MPD_leak_1=" "
                         then do MPD_leak_1=".";end;
                              MPD_L1=MPD_leak_1+0;


if MPD_leak_2='Y' or MPD_leak_2='y'
     then do MPD_leak_2=1;end;
          else if MPD_leak_2="N" or MPD_leak_2="n"
               then do MPD_leak_2=0;end;
                    else if MPD_leak_2=" "
                         then do MPD_leak_2=".";end;
                              MPD_L2=MPD_leak_2+0;
```

```
if Pancreatic_atrophy_1='Y' or Pancreatic_atrophy_1='y'
     then do Pancreatic_atrophy_1=1;end;
          else if Pancreatic_atrophy_1="N" or Pancreatic_atrophy_1="n"
               then do Pancreatic_atrophy_1=0;end;
                    else if Pancreatic_atrophy_1=" "
                         then do Pancreatic_atrophy_1=".";end;
                              PA1=Pancreatic_atrophy_1+0;

if Pancreatic_atrophy_2='Y' or Pancreatic_atrophy_2='y'
     then do Pancreatic_atrophy_2=1;end;
          else if Pancreatic_atrophy_2="N" or Pancreatic_atrophy_2="n"
               then do Pancreatic_atrophy_2=0;end;
                    else if Pancreatic_atrophy_2=" "
                         then do Pancreatic_atrophy_2=".";end;
                              PA2=Pancreatic_atrophy_2+0;

if Contour_abnormality_of_bile_duct='Y' or Contour_abnormality_of_bile_duct='y'
     then do Contour_abnormality_of_bile_duct=1;end;
          else if Contour_abnormality_of_bile_duct="N" or
          Contour_abnormality_of_bile_duct="n"
               then do Contour_abnormality_of_bile_duct=0;end;
                    else if Contour_abnormality_of_bile_duct=" "
                         then do Contour_abnormality_of_bile_duct=".";end;
                              CABD1=Contour_abnormality_of_bile_duct+0;

if Contour_abnormality_of_bile_duc0='Y' or Contour_abnormality_of_bile_duc0='y'
     then do Contour_abnormality_of_bile_duc0=1;end;
               else if Contour_abnormality_of_bile_duc0="N" or
               Contour_abnormality_of_bile_duc0="n"
                    then do Contour_abnormality_of_bile_duc0=0;end;
```

```
                              else if Contour_abnormality_of_bile_duc0=" "
                                   then do
                              Contour_abnormality_of_bile_duc0=".";end;

CABD2=Contour_abnormality_of_bile_duc0+0;


if Bile_duct_stricture_1='Y' or Bile_duct_stricture_1='y'
     then do Bile_duct_stricture_1=1;end;
          else if Bile_duct_stricture_1="N" or Bile_duct_stricture_1="n"
               then do Bile_duct_stricture_1=0;end;
                    else if Bile_duct_stricture_1=" "
                         then do Bile_duct_stricture_1=".";end;
                              BDS1=Bile_duct_stricture_1+0;

if Bile_duct_stricture_2='Y' or Bile_duct_stricture_2='y'
     then do Bile_duct_stricture_2=1;end;
          else if Bile_duct_stricture_2="N" or Bile_duct_stricture_2="n"
               then do Bile_duct_stricture_2=0;end;
                    else if Bile_duct_stricture_2=" "
                         then do Bile_duct_stricture_2=".";end;

BDS2=Bile_duct_stricture_2+0;

SBS1=Side_branch_size__mm__1;
SBS2=Side_branch_size__mm__2;

MPD_size1=MPD_size__mm__1;
MPD_size2=MPD_size__mm__2;

cstage1=CAMBRIDGE_STAGE_1;
```

```
     cstage2=CAMBRIDGE_STAGE_2;




     drop SBE_3_1 SBE___3_1 SBE___3_2 SBE___3_20 MPD_irregularity_1
     MPD_irregularity_2
          MPD_calculi_1 MPD_calculi_2 Pseudocyst_1 Pseudocyst_2 MPD_leak_1
          MPD_leak_2 Pancreatic_atrophy_1 Pancreatic_atrophy_2
          Contour_abnormality_of_bile_duct Contour_abnormality_of_bile_duc0
          Bile_duct_stricture_1 Bile_duct_stricture_2 Side_branch_size__mm__1
          Side_branch_size__mm__2 MPD_size__mm__1 MPD_size__mm__2 CAMBRIDGE_STAGE_1
          CAMBRIDGE_STAGE_2 MPD_stricture_1 MPD_stricture_2
          Duration_of_symptoms_days_ Duration_of_symptoms
          Duration_of_symptoms_years_
          Time_to_DM_years_  Time_to_Exo_failure;
run;


data bimal12_15p2;
     set dat.bimal12_15p2;


     SBE_L31=SBE_3_1+0;
     SBE_G31=SBE___3_1+0;
```

```
SBE_L32=SBE___3_2+0;
SBE_G32=SBE___3_20+0;


MPD_i1=MPD_irregularity_1+0;* converting character to numeric;

MPD_i2=MPD_irregularity_2+0;* converting character to numeric;

MPD_s1=MPD_stricture_1+0;* converting character to numeric;


MPD_s2=MPD_stricture_2+0;* converting character to numeric;

MPD_c1=MPD_calculi_1+0;

MPD_c2=MPD_calculi_2+0;

Psc1=Pseudocyst_1+0;

Psc2=Pseudocyst_2+0;

MPD_L1=MPD_leak_1+0;

MPD_L2=MPD_leak_2+0;


PA1=Pancreatic_atrophy_1+0;

PA2=Pancreatic_atrophy_2+0;
```

```
     CABD1=Contour_abnormality_of_bile_duct+0;

     CABD2=Contour_abnormality_of_bile_duc0+0;


     BDS1=Bile_duct_stricture_1+0;

     BDS2=Bile_duct_stricture_2+0;

     SBS1=Side_branch_size__mm__1;
     SBS2=Side_branch_size__mm__2;

     MPD_size1=MPD_size__mm__1;
     MPD_size2=MPD_size__mm__2;

     cstage1=CAMBRIDGE_STAGE_1;
     cstage2=CAMBRIDGE_STAGE_2;



     drop SBE_3_1 SBE___3_1 SBE___3_2 SBE___3_20 MPD_irregularity_1
MPD_irregularity_2
          MPD_calculi_1 MPD_calculi_2 Pseudocyst_1 Pseudocyst_2 MPD_leak_1
          MPD_leak_2 Pancreatic_atrophy_1 Pancreatic_atrophy_2
          Contour_abnormality_of_bile_duct Contour_abnormality_of_bile_duc0
          Bile_duct_stricture_1 Bile_duct_stricture_2 Side_branch_size__mm__1
          Side_branch_size__mm__2 MPD_size__mm__1 MPD_size__mm__2 CAMBRIDGE_STAGE_1
          CAMBRIDGE_STAGE_2 MPD_stricture_1 MPD_stricture_2
          Duration_of_symptoms_days_ Duration_of_symptoms
          Duration_of_symptoms_years_
          Time_to_DM_years_  Time_to_Exo_failure;
     run;


data bimal12_15;
     set bimal12_15p1 bimal12_15p2;
     MRI_date=Date_of_Ist_MRI;
     DM_date=Date_of_onset_of_DM;
     EXO_date=Date_of_Exocrine_failure_yrs_;

     bsline_DM=DM_date-MRI_date;
     bsline_EXO=EXO_date-MRI_date;

     format MRI_date MONYY5. DM_date MONYY5. EXO_date MONYY5.;

     if bsline_DM>0 or bsline_DM='.' then do bsline_DM=0;end;
     else do bsline_DM=1;end;

     if bsline_EXO>0 or bsline_EXO='.' then do bsline_EXO=0;end;
     else do bsline_EXO=1;end;

     lookdate='01DEC08'd;
     format lookdate MONYY5.;

     DM_now=lookdate-DM_date;
     if DM_now='.' then do DMnow=0;end; else do DMnow=1;end;

     EXO_now=lookdate-EXO_date;
     if EXO_now='.' then do EXOnow=0;end; else do EXOnow=1;end;

      Psc_now=0;
      BDS_now=0;

     if Psc1=1 or Psc2=1 then do Psc_now=1;end;

     if BDS1=1 or BDS2=1 then do BDS_now=1;end;


     bsl_scr=bsline_EXO+bsline_DM+Psc1+ BDS1;

     flu_scr=DMnow+ EXOnow+Psc_now+BDS_now;
     any1=flu_scr>1;

run;


data fluscore;
     set bimal12_15;

     sbs=SBS1;
     ylowmin=2; yhimin=3;     g1min=0.15;
     a1min=(yhimin+ylowmin)/2;
     b1min=(yhimin-ylowmin)/(2*log((1-g1min)/g1min));
     d12s=(1+exp((sbs-a1min)/b1min))**-1; * Smaller -the-better;



     logit_95=log((.95)/(1-.95));
     logit_08=log((.8)/(1-.8));

     m=-(logit_95-logit_08)/3;
     dt1=1/(1+exp(-(4+m*MPD_size1)));
```

```
ylowmin1=2; yhimin1=2.5;      g1min1=0.2;
a1min1=(yhimin1+ylowmin1)/2;
b1min1=(yhimin1-ylowmin1)/(2*log((1-g1min1)/g1min1));
dt2=(1+exp(-(MPD_size1-a1min1)/b1min1))**-1;* bigger-the-better;

d13=dt1*dt2;


If PA1=1 then do d5=1;end;
If PA1=0 then do d5=0;end;


if MPD_s1=1 then do d9=0.5;end;
if MPD_s1=0 then do d9=1;end;

if MPD_l1=1 then do d8=0.5;end;
if MPD_l1=0 then do d8=1;end;

if concomittant_alcohol_use =1 then do d1=0.5;end;
if concomittant_alcohol_use =0 then do d1=1;end;

if ongoing_smoking=1 then do d2=1;end;
if ongoing_smoking=0 then do d2=0.5;end;

if SBE_G31=1 then do d4=0.5;end;
if SBE_G31=0 then do d4=1;end;

if MPD_i1=1 then do d3=0.5;end;
if MPD_i1=0 then do d3=1;end;
```

```
     if MPD_c1=1 then do d6=0.5;end;
     if MPD_c1=0 then do d6=1;end;

     if CABD1=1 then do d11=0.5;end;
     if CABD1=0 then do d11=1;end;


     ds13=(d12s*d8*d9*d1*d11*d4*d3*d6*d13)**(1/9);
     dnew=ds13*10;
     dnew=ds13;

run;

/***********************************************************************************
****************/


proc iml;
     use fluscore;
     read all var{dnew} into x;
     read all var{flu_scr} into R;
     read all var{d12s} into d1;
     read all var{d8} into d2;
     read all var{d9} into d3;
     read all var{d1} into d4;
     read all var{d11} into d5;
     read all var{d4} into d6;
     read all var{d3} into d7;
     read all var{d6} into d8;
     read all var{d13} into d9;
```

```
start initial;

     alpha={2.2397 0.9029 -1.2476 -3.4281}; beta=-0.1647;

     a1=alpha[,1];
     a2=alpha[,2];
     a3=alpha[,3];
     a4=alpha[,4];


     w1=1; w2=1; w3=1; w4=1; w5=1; w6=1; w7=1; w8=1; w9=1;
     wsum=w1+w2+w3+w4+w5+w6+w7+w8+w9;
     w=(w1//w2//w3//w4//w5//w6//w7//w8//w9);

     N=nrow(x);

finish initial;
run initial;


start ll(abeta)
     global(d1,d2,d3,d4,d5,d6,d7,d8,d9,w1,w2,w3,w4,w5,w6,w7,w8,w9,
     wsum,p1,p2,p3,p4,R);


     S=((d1##w1)#(d2##w2)#(d3##w3)#(d4##w4)#(d5##w5)#(d6##w6)#(d7##w7)#(d8##d8)
     #(
     d9##w9))##(1/wsum);
```

```
          S_scale=S;

          *Pi’s for the likelihood equations;
          p4=(exp(abeta[4]+abeta[5]*S_scale))/(1+(exp(abeta[4]+abeta[5]*S_scale)));
          p3=(exp(abeta[3]+abeta[5]*S_scale))/(1+(exp(abeta[3]+abeta[5]*S_scale)));
          p2=(exp(abeta[2]+abeta[5]*S_scale))/(1+(exp(abeta[2]+abeta[5]*S_scale)));
          p1=(exp(abeta[1]+abeta[5]*S_scale))/(1+(exp(abeta[1]+abeta[5]*S_scale)));
          pi4=p4;*P(Y>4);
          pi3=p3-p4;*P(Y>3)-P(Y>4);
          pi2=p2-p3;*P(Y>2)-P(Y>3);
          pi1=p1-p2;*P(Y>1)-P(Y>2);
          pi0=1-p1;


     sum=(R=0)#log(pi0)+(R=1)#log(pi1)+(R=2)#log(pi2)+(R=3)#log(pi3)+(R=4)#log(pi4);
          sum_sum=sum(sum);
          m2lik=-2*sum_sum;*-2LL;
          return(sum_sum);
     finish ll;


/***********************************************************************************
*****
Starting Values
* starting values should be a row vector;
************************************************************************************
****/

beta0={2.2397, 0.9029, -1.2476, -3.4281,-0.1647};
```

```
/***********************************************************************************
***
  Options
************************************************************************************
***/
optn=j(1,11,.);
optn[1]=1; *min=0 max=1;
optn[2]=0; *Controls the amount of printout;

/***********************************************************************************
***
Termination Criteria
************************************************************************************
***/
tc=j(1,13,.);
tc[1]=5000; *maximum iterations;


/***********************************************************************************
**
     Control parameter vector
************************************************************************************
**/
par=j(1,10,.);
par[2]=1E-1; * initial step length;
par[6]=0.05;*required accuracy of the line search;

/**********************************
Call procedure
**********************************/
```

```
start varcov;
     call nlpnrr(rc, abeta, 'll', beta0, optn);
     call nlpfdd(f,g,h,"ll",abeta);
     var=inv(-h);
     se=sqrt(vecdiag(var));
     gvar=det(var);
     dopt=gvar;
finish varcov;

run varcov;



%include simplex;

start function;
     w1=parms[1];w2=parms[2];w3=parms[3];w4=parms[4];w5=parms[5];w6=parms[6];w7=parm
s[7];
     w8=parms[8]; w9=9-w1-w2-w3-w4-w5-w6-w7-w8;
     w=(w1//w2//w3//w4//w5//w6//w7//w8);
     ylowmax=0.1; yhimax=0.2; gmax=0.1;
     amax=(ylowmax+yhimax)/2;
     bmax=(yhimax-ylowmax)/(2*log((1-gmax)/gmax));
     ylowmin=5.5; yhimin=7.5; gmin=0.1;
     amin=(ylowmin+yhimin)/2;
     bmin=(yhimin-ylowmin)/(2*log((1-gmin)/gmin));


     minw=min(w1,w2,w3,w4,w5,w6,w7,w8,w9);
     maxw=min(w1,w2,w3,w4,w5,w6,w7,w8,w9);
     dmax=(1+exp(-((minw-amax)/bmax)))**-1;
     dmin=(1+exp(((maxw-amin)/bmin)))**-1;
     dt=(dmax*dmin)**(1/2);


     if((w1<0)+(w2<0)+(w3<0)+(w4<0)+( w5<0)+( w6<0)+( w7<0)+( w8<0)+( w9<0)
     +((w1+w2+w3+w4+w5+w6+w7+w8+w9)>9) )>0 then fn_value=10**30;

          else do;

               run varcov;
               scalefactor=.000066;
               constant=lambda*scalefactor;
               desterm = constant*(1-dt);
               fn_value = dopt + desterm;

          end;
finish;

Start Optima;
     in_parms=(w1//w2//w3//w4//w5//w6//w7//w8//w9);
     in_steps=in_parms*.5;

     run varcov;
     se_betas = sqrt(vecdiag(var));

     run simplex;
     run function;


     se_betas = sqrt(vecdiag(var));
```

```
finish;


start grid;
     design=0;


     *do lambda=0 to .05 by 0.01;
     lambda=0.5;

      do initw1=1 to 2 by .5;
          do initw2=1 to 1 by .5;
               do initw3=0.5 to 1 by .5;
                    do initw4=0.5 to 1 by .5;
                         do initw5=.5 to 1 by .5;
                              do initw6=0.5 to 1 by .5;
                                   do initw7=.5 to 1 by .5;
                                    initw8=1.5;* to 1 by .5;
                                         initw9=9-initw1-initw2-initw3-initw4-
          initw5-
                                         initw6-initw7-initw8;


                                        w1=initw1; w2=initw2; w3=initw3;w4=initw4;
                                        w5=initw5; w6=initw6; w7=initw7;w8=initw8;
                                        w9=initw9;
                                        w=(w1||w2||w3||w4||w5||w6||w7||w8||w9)`;
                                        run optima;
                                        design=design+1;
                                        labels={'design' 'initw1' 'initw2' 'initw3'
                                        'initw4' 'initw5' 'initw6' 'initw7' 'initw8'
                                        'initw9''dopt' 'lambda' 'constant' 'count'
                                   'w1'
                                        'w2' 'w3' 'w4' 'w5' 'w6' 'w7' 'w8' 'w9' 'dt'
                                        'determ' 'fn_value' };
                                        results= results//(design||in_parms[1]||
                                        in_parms[2]||in_parms[3]||in_parms[4]||
                                        in_parms[5]||in_parms[6]||in_parms[7]||
                                        in_parms[8]||in_parms[9]||dopt||lambda||
                                        constant||count||w1||w2||w3||w4||w5||w6||w7|
                                        |w8||w9||dt||determ||fn_value);

                                   end;
                              end;
                         end;
                    end;
               end;
          end;
     end;

     create results from results[colname=labels]; append from results;
finish;
run grid;

quit;


proc sort data=results; by lambda fn_value;run;
proc print data=results;by lambda;run;

data final;
```

```
     set results; by lambda;

proc print;

     goptions ftext=script htext=1.8;
     symbol1 i=join l=1 c=blue;
     symbol2 i=join l=3 c=red;

     axis1 label=(a=90 font=simulate height=1.5 'D (solid line, blue)')
     value=(font=simulate height=1.3);
     axis2 label=(a=270 font=simulate height=1.5 'Generalized Variance (dashed
     line,red)' ) value=(font=simulate height=1.3);

proc gplot;
   plot dt*lambda/vaxis=axis1 ;
   plot2 dopt*lambda/vaxis=axis2;
   label lambda='l';
   title ' ';
run;
quit;
```

VITA

Rhonda Ellis was born on May 26, 1978, in Williamsburg, Virginia. She graduated from Bruton High school in Williamsburg in June of 1996. Rhonda then graduated from Hampton University in Hampton, Virginia with a Bachelor of Science degree in August, 2001. She remained at Hampton University and obtained a Master of Science degree in Applied Mathematics in May, 2003. Rhonda then began a doctoral program in Biostatistics at Virginia Commonwealth University in Richmond, Virginia. Rhonda has accepted a position as Assistant Professor with the Department of Mathematics at Norfolk State University, in Norfolk Virginia, upon completion of her doctoral program.