12-2009

An Investigation on the Use and Flexibility of Genetic Algorithms for Logistic Regression

Sara Yoder
Clemson University, bethy@clemson.edu

AN INVESTIGATION ON THE USE AND FLEXIBILITY OF GENETIC ALGORITHMS FOR LOGISTIC REGRESSION

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Industrial Engineering

---

by
Sara Elizabeth Yoder
December 2009

Accepted by:
Dr. Mary Elizabeth Kurz, Committee Chair
Dr. Robert Brookover
Dr. Byung Rae Cho
Dr. Scott Shappell

ABSTRACT

Social scientists and other users of large data sets often desire a model to predict the probability that some condition exists, such as the probability that a person has diabetes or that a credit card transaction will be fraudulent. In general, this can be done by data mining techniques, which allow multiple records of data composed of numerous independent variables and one dependent variable to be examined in a statistical fashion to make a predictive model. One particular technique used is logistic regression. Logistic regression forms a predictive model based on a set of independent variables by assigning coefficients to these variables to maximize a non-linear function. The state-of-the-art for creating logistic regression models requires the modeler to select independent variables and the use of an iterative search technique to solve the underlying non-linear optimization.

This dissertation investigates the use of genetic algorithms for creating a logistic regression model. The use of this optimization technique facilitates resolution of two critiques of the state-of-the-art: (1) user selection of independent variables allows bias from the user to enter into the logistic regression model; (2) the iterative optimization method used can result in sub-optimal models being accepted. The use of genetic algorithms in the place of the current optimization technique effectively addresses these concerns.

Data of increasing complexity are considered, from one to several independent variables. In response, genetic algorithms that allow for increasing flexibility are developed. Through extensive computational studies, a robust genetic algorithm that

allows for selection of independent variables and setting of parameter values is developed.

The power of the developed approach is demonstrated in a case study of general aviation accident data with five hundred cases and thirteen independent variables.

DEDICATION

To my father, Joseph Stasiukaitis, who planted the seed. To my mother, Sara Stasiukaitis, who continuously tended to the plant with loving care that only a mother can give. Both of my parents have instilled in me an appreciation for education and a desire to continue learning throughout life, for which I am forever grateful. I hope to do the same for others.

To my family: Bryan, Gibson, and Cannon Yoder. Without their sacrifices, love, and support, it would not have been possible for me to complete this journey.

ACKNOWLEDGMENTS

I will be forever grateful for the opportunity to work with my advisor, Dr. Mary Beth Kurz. Her support and patience are unrivaled. I am thankful for being molded by her knowledge and advice. Throughout this long journey, she graciously allowed the role of advisor to encompass more than just academic advice; for that I am greatly appreciative. My years as a graduate student have been much more enjoyable because of Dr. Kurz.

I would like to acknowledge the members of my dissertation committee, Dr. Robert Brookover, Dr. Byung Rae Cho, and Dr. Scott Shappell. I am thankful for their support and suggestions that have added to this work.

I want to thank my colleagues, mentors, and dear friends, Drs. Mark and Judith McKnew. I cherish their advice and ability to always have the right thing to say. Without them, I would not be the person I am today.

TABLE OF CONTENTS

Page

Table of Contents (Continued)

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER ONE

1 INTRODUCTION AND LITERATURE REVIEW

1.1 **Logistic Regression**

Logistic regression, a form of the general linear regression model, is used with data having a dependent variable, also referred to as a response variable, that is dichotomous (taking on one of two values) in a number of different application areas (Agresti 1990). In the area of human factors research, it has been used in several papers concerning accidents – the determination of whether an accident is fatal or not (Al-Ghamdi 2002) and predicting both active and passive crashes (Mesken, *et al*. 2002). Logistic regression has also been used to predict falls (Hanson, *et al*. 1999) and the outcome of a CDL (Commercial Driving License) driving test (Carnahan, Meyer, Kuntz 2003). Any application concerned with success/failure or presence/absence lends itself to using the logistic regression model.

Logistic regression is very useful in that it makes no assumptions about the independent variables. There is no requirement of the data being drawn from a normal distribution. The variance of each group does not need to be equal. The independent variables can be discrete, continuous, or categorical, or a combination of these (Tabachnik and Fidell 2001). This flexibility allows logistic regression to be used in a variety of applications.

Given a set of data representing the values of the independent variables and the corresponding response variable, estimates of the parameters can be made. In general,

the probability that the response variable Y has value 1 is expressed as a function of the parameters $\beta_0, \beta_1, \ldots, \beta_p$ and the values of the p independent variables $x_1, x_2, \ldots, x_p$(see equation 1). Equation 1 includes two mathematically equivalent forms of the logistic regression model. This dissertation will favor the second version.

$$\Pr\left(Y=1 \mid x_1, x_2, \ldots, x_p\right) = \frac{e^{\beta_0+\sum_{i=1}^{p}\beta_i x_i}}{1+e^{\beta_0+\sum_{i=1}^{p}\beta_i x_i}} = \frac{1}{1+e^{-\left(\beta_0+\sum_{i=1}^{p}\beta_i x_i\right)}} \quad (1)$$

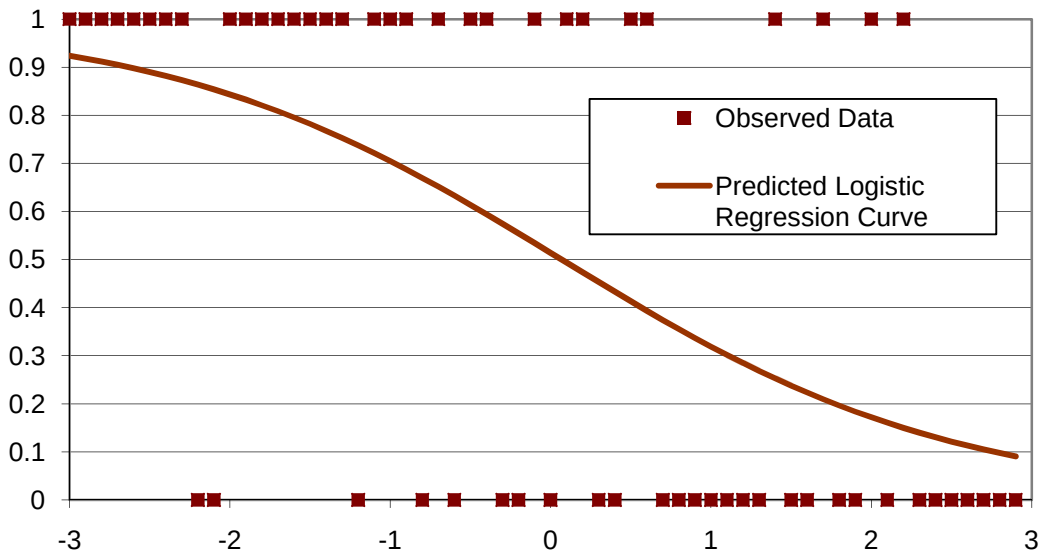

Figure 1: Depiction of Logistic Regression Model and Sample Data

Figure 1 shows an example of a set of data with one independent variable and a possible logistic regression model. The observed data shows the y values for each of the input x values, indicating, for example, that when x is -3, y is 1 and when x is 0, y is 0. This figure demonstrates that while the y values for the input data are either 0 or 1, the

resulting logistic regression model returns a real value between 0 and 1 indicating the probability that the dependent variable is 1 based on the corresponding independent variable.

In developing a logistic regression model, a modeler usually determines which independent variables from a given set should be included in the model. As seen in Foster, *et al.* (2006), there are three approaches: direct, sequential, and stepwise. The direct method requires all of the independent variables to be included in the model. It is used when there is no preconceived hypothesis about the importance or order of the independent variables. If there is some hypothesis about the importance of the independent variables, then the sequential method should be used. In this method, the modeler chooses the order of importance such that the independent variable that has the most priority gets entered into the model first. Then other variables are added in order of importance and assessed to determine if they have improved the model. Finally, the stepwise logistic regression method allows the model to be assessed as it is being built. As an independent variable is added to the model, it is tested to see if it has improved the predictive ability of the model measured by one of the following: F, p, or R^2 values. If so, it is added to the model and if not, it is excluded from the model. One of the challenges is to choose independent variables that are correlated to the dependent variable, but not highly correlated to other independent variables (for instance a correlation less than 0.7). The predictive power of an independent variable can be masked by a previously included independent variable if the two are correlated.

A software package, such as SAS or SPSS, is usually used to estimate the parameters of the model and determine how well the model fits the set of data. The modeler can often set certain criteria to determine if the model is good enough, but one must keep in mind that the model's quality depends on the order in which the independent variables are selected. We hypothesize that a modeler's preconceived notions of variable importance may taint this process. In consideration of this issue, we intend to provide the modeler with tools that create models with sets of independent variables that are not user-specified.

The algorithms used by software packages introduce additional concerns regarding the resultant models. Software packages also use iterative methods to acquire the maximum likelihood estimates of the coefficients. The covariance matrix of the estimators is usually approximated. Furthermore, any criterion based on the variances of the estimators depends on the unknown values of the parameters (Russell, *et al.*, 2009). Likewise, the use of matrix inversion in these iterative methods can limit the number of predictor variables that may be used (Genkin, Lewis, and Madigan, 2007). Therefore, due to both user bias and inherent mathematical issues with numerical stability, we propose the use of genetic algorithms to both select predictor variables and determine their parameters.

1.2 **Genetic Algorithms**

Genetic algorithms are one of many heuristic optimization methods that mimic the biological process of evolution. An initial population is created. "Parents" are selected and "children" are produced. The next generation is created by keeping some of the best

children from the previous generation and constructing the rest by genetic operators which vary depending on the design (Goldberg 1989). The overall scheme is shown in Figure 2.

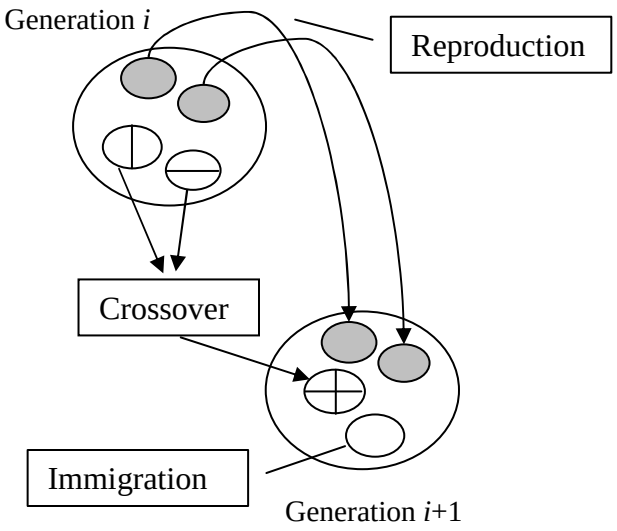

Figure 2: A Basic GA Scheme

In general, genetic algorithms randomly generate a population of chromosomes. Chromosomes are composed of series of numbers. In light of the genetic analogy these numbers are referred to as “genes”. Chromosomes are decoded into problem solutions. Every chromosome is decoded into a feasible solution which is then evaluated to find the associated objective function value. The best of the current population is recorded as the incumbent. A new population, which possibly results in a better “incumbent”, is formed through the application of genetic operators which depend on the problem. For example,

uniform crossover chooses two parents from the population randomly. For each gene of the new child chromosome, a parent is chosen randomly as the contributor of that gene. Immigration can be used to introduce variation into the population by randomly generating a new chromosome. After the new generation has been filled, its chromosomes are decoded into solutions, the objective function value is calculated, and the stopping criteria are considered.

Genetic algorithms (GAs) have been used in many areas, including job rotation scheduling (Carnahan, *et al*. 2000) and predicting dwelling fire occurrence (Yang, *et al*. 2006). Carnahan, Meyer, and Kuntz (2003) also used GAs to predict the outcome (pass/fail) of a CDL driving test.

In light of previous research, we select genetic algorithms as the tool by which various models will be created. Initially, a genetic algorithm will be used to determine the parameters of the logistic regression model when certain independent variables are in the model. That algorithm will then be altered so that it will select certain independent variables from set of possible independent variables that should be included in the logistic regression model and determine the parameters for the model.

1.2.1 **Selected genetic operators**

Genetic operators are processes that change the current population of chromosomes. Over time, these processes help lead the genetic algorithm to find a better solution by exploring the solution space. Different operators affect chromosomes in different ways. A few operators are discussed here.

- Crossover: This process first selects two chromosomes to be the parents and then we create at least one child from those parents. Selecting parents is generally done by one of two mechanisms: (1) the simplest way to select parents is randomly or (2) with a roulette wheel selection, in which each chromosome in a generation is assigned a probability proportional to its relative quality in the generation. A random number is generated and then the appropriate chromosome is selected as a parent.

  In SPC (single point crossover), the crossover point is randomly selected, i.e. after the second gene. The first child chromosome then receives its first two genes from parent 1 and the rest from parent 2; the second child receives the first two genes from parent 2 and the rest from parent 1. Figure 3 below illustrates how child 1 and child 2 are created from parent 1 and parent 2.

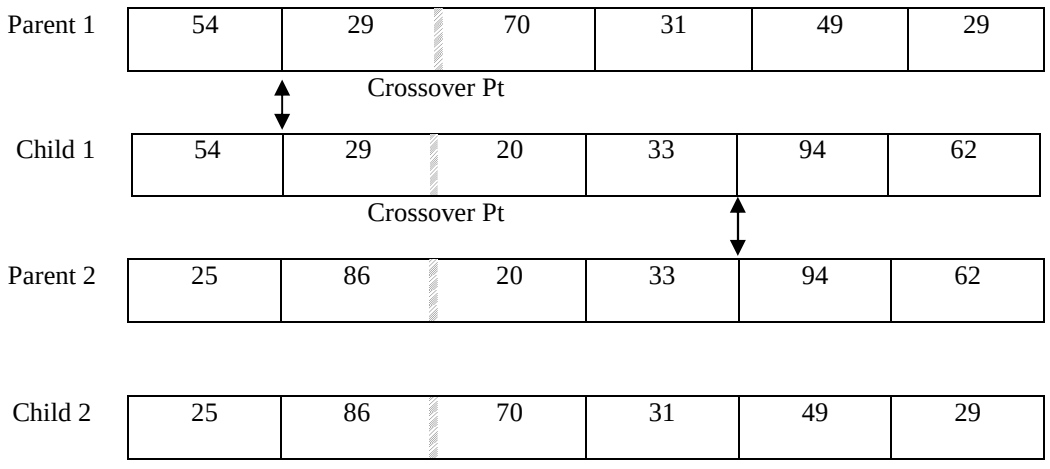

Figure 3: Formation of Children From SPC

In TPC, two point crossover, the two children are created in a similar fashion as that in single point crossover except there are two places where the genes are swapped from parent to child. Child 1 would receive the genes from parent 1 that come before the first crossover point and those that come after the second crossover point. Child 1 will receive the genes that come between the two crossover points from parent 2. Child 2 will be created in the opposite manner. Figure 4 illustrates the formation of child 1 and child 2.

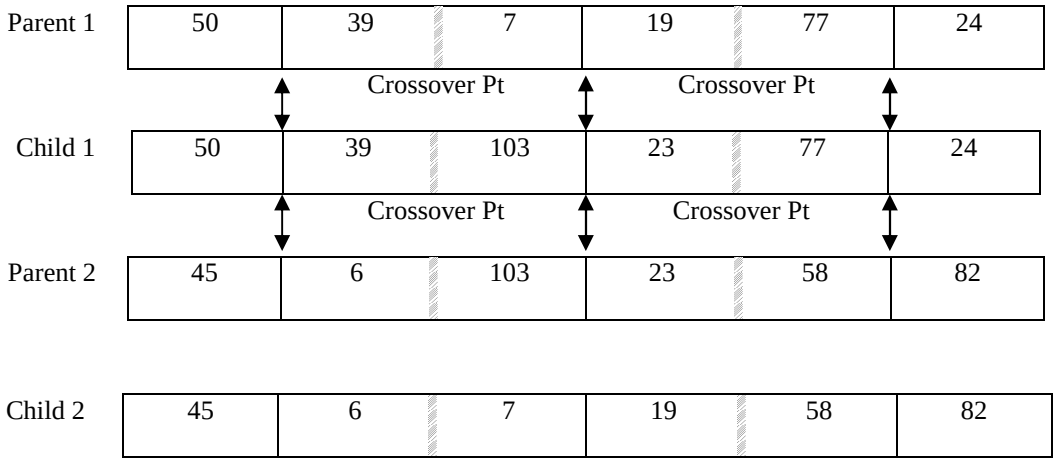

Figure 4: Formation of Children From TPC

In uniform crossover, each gene of the child has an equal probability of being selected from parent 1 and of being selected from parent 2. In parametric uniform crossover (PUC), each gene of the child has a set probability of being selected from parent 1, with the complementary probability of being selected from parent 2.

Sometimes, a tournament selection process is used to choose which of the two children will enter the next generation. The child with the better objective function value moves to the next generation.

- Mutation: With the genetic operator of mutation, each gene in a chromosome with some low given probability gets changed. The change depends on the representation of the chromosome. For example, with a binary representation (0 or 1), a change would simply consist of "flipping" the gene so that a zero would become a one or vice versa.
- Immigration: In immigration, a specified percentage of the new generation is created by forming new chromosomes from the initialization process used in the first generation.

1.3 **Appropriateness of Genetic Algorithms for Logistic Regression**

Logistic regression is a hard, non-linear optimization problem. As discussed by Goldberg , the use of genetic algorithms is well suited to hard optimization problems (1989). The literature review in the next section highlights previous uses of genetic algorithms for logistic regression and identifies gaps in the literature.

1.4 **Literature Review**

In this section, we discuss some specific papers involving genetic algorithms and logistic regression that can provide insight for our proposed work.

•Pasia, Hermosilla, and Ombao (2005)

Pasia, Hermosilla, and Ombao compared the use of a genetic algorithm with three classical numerical optimization methods, one being logistic regression, in parameter estimation. The least squares method was used for the parameter estimation.
The six stages to their GA are discussed: genetic representation, initialization, function evaluation, creation of new population, test, and loop. The genetic representation is usually a binary representation, but can be encoded in other ways, depending on the nature of the problem. In the initialization phase, a population size must be determined and then a set of chromosomes need to be randomly created. The population size and the initial population can both affect the genetic algorithm's performance. In function evaluation, each chromosome is evaluated based on some predefined fitness criterion, usually the objective function. The creation of a new population incorporates several steps. Two parent chromosomes must be selected (there are many ways to do this); the chance of being selected increases as the fitness increases. Then a crossover technique is employed to create a new offspring chromosome (again, there are many ways to do this) with some probability. And with a very small probability (< 0.1), some chromosomes are mutated (small changes to one chromosome). Finally, the new population is created by the acceptance of these offspring; this new population now becomes the population on which the GA currently works. A test is conducted to see if the GA is "done"; it is if the defined stopping criteria have been met. There are several stopping criteria that are commonly utilized. The loop back to the function evaluation will occur if the GA is not done and these processes will be repeated until the stopping criteria are met.

Pasia, *et al.*'s GA had a population size of 100. Each chromosome was composed of two vectors, the first being a real number between -1 and 1 and the second being an integer from the set {10, 20, 30, 40, 50}. A binary tournament was used to select parents and a modified uniform crossover was used to create the offspring. The offspring with the highest fitness was kept. The mutation rate was .10 and the stopping criterion was 100 generations. This constituted one run; five runs were performed on each data set. For the logistic regression model, 500 data sets were created with each containing 60 observations using a logistic model with a constant and one independent variable. The parameters for the model were set at $\beta_1 = 1$ and $\beta_2 = -1$ and the 60 observations for x were -3.0. -2.9, … , 2.8, 2.9. Four classical methods were used with the software SAS to get the parameter estimates. Comparable results were reported for the genetic algorithm and SAS.

This paper provides a benchmark genetic algorithm for a small and specific logistic regression application which will allow initial experimentation. The details of Pasia, *et al.*'s procedures and results will be discussed thoroughly in Chapter Three of this paper.

•Russell, Eccleston, Lewis, and Woods (2009)

Russell, *et al*. address the issues of using Maximum Likelihood estimators for the model parameters in a logistic regression model. The bias that is created by using iterative methods to get the ML estimators is discussed. The main concern is for small experiments with limited observations. They experiment with a logistic regression model that has a single variable and two parameters. They compare the ML estimators and the

Maximum Penalized Likelihood estimators of β_0 and β_1. The authors then offer an alternative method for obtaining the estimates for the parameters.

This paper provides support for considering an alternative method for parameter estimation for logistic regression, rather than software packages that are readily available.

•Genkin, Lewis, and Madigan (2007)

Genkin, Lewis, and Madigan discuss a few shortcomings of maximum likelihood estimation used in conjunction with logistic regression parameter estimates. Their specific area of interest is with "short, fat" data sets – those that have a large number of predictor variables, usually exceeding the number of observations. They address the problem of computing maximum likelihood estimates on software packages, due to the use of matrix inversion. Additionally, they point out that using the maximum likelihood method will often overfit the data with these data sets. Genkin, *et al*. suggest a Bayesian approach to logistic regression that avoids these problems.

This paper addresses computational problems and provides direction for future work.

•Mesken, Lajunen, and Summala (2002)

Mesken, Lajunen, and Summala examined 1126 Finnish drivers' responses to the Driver Behaviour Questionnaire (DBQ). They were interested in the relationship between aggressive violations and accident risk and if age and gender had an effect on them. They also sought to replicate a previous experiment using Finnish drivers.

Factor analysis was performed using four factors: lapses, errors, interpersonal violations, and speeding violations. Correlations were calculated for the factors and variables such as age, gender, and yearly mileage driven. Analysis of variance was used to look at gender, age, and the four DBQ scales.

Mesken, *et al*. wanted to examine how errors, lapses, and the two types of violations were related to accidents and traffic penalties. Variables for number of accidents and number of penalties were assigned 1 if the driver had 1 or more in the past year and 0 if not. Logistic regression was run with age, gender and mileage being forced into the model on the first run. Then the scores for the DBQ scales, errors, lapses, and the two violations were entered into the models according to the stepwise method.

Age and DBQ errors predicted active accidents. Age, mileage, and interpersonal violations predicted passive accidents. Young drivers were more often involved in both types of crashes. Age and mileage were predictors of speeding tickets. The authors found results that were consistent with previous findings.

This paper considers the actions of the drivers and different variables that affect them. We will look at pilots and various factors that may affect the outcome of an aviation accident.

•Al-Ghamdi (2002)

Al-Ghamdi collected data from 560 drivers that filed an accident report with the traffic police in Riyadh, Saudi Arabia. Only accidents that occurred on urban roads and that involved an injury of some sort (including a fatality) were included. Ten variables

were derived from the accident reports and the response variable of fatal or non-fatal accident was used. Of the nine independent variables, one (age) was continuous; the other eight were categorical. Preliminary hypothesis tests of proportions were conducted to reduce the number of categories in those eight variables.

Logistic regression was conducted using the backward selection process. The nine independent variables were reduced to two, based on a significance level of 5%. These were the location of the accident and the cause of the accident (i.e. run red light). Then interaction between these two variables was introduced into the model and found to be significant.

Age was investigated further even though it was not shown to be significant in the model. This decision was based on previous findings. A quadratic term for age was introduced into the model with the two significant variables, but was found to be not significant at the 5% level.

Al-Ghamdi concluded that for an accident that occurs at a non-intersection location the odds of the accident being fatal are 2.64 higher than for those at an intersection. Running a red light will have odds that are 2.72 times higher for a fatal accident than for accidents that are not related to running a red light. He also concluded for non-intersection locations the odds of being in a fatal crash in a wrong-way accident are three times higher than that of being in a failure-to-yield accident. In fact, he concluded that non-intersection accidents are more serious than intersection accidents. Wrong-way related accidents were found to be more likely to be fatal than accidents with other causes.

This paper provides us with an example of logistic regression used to predict fatal or non-fatal crashes. While it considers automobile crashes, our data of aviation crash outcomes is similar and will be used in the same manner. Our data set will be comprised of accident report data also.

•Carnahan, Meyer, and Kuntz (2003)

Carnahan, Meyer, and Kuntz compared the traditional statistical method of logistic regression to a machine learning method of genetic algorithms. The study was conducted to predict whether a driver trainee would pass the CDL test the first time based on ten independent variables (scores accumulated during the driving training course). Data was collected from 37 trainees; a set of 10 cases (5 pass, 5 fail) were the test set and the remaining 27 cases were the training set. The training set was used to create the model and then the model was validated using the test set.

Equation (2), representing a variant of the standard logistic regression model, was used for the logistic regression model, where *Y* is the probability that the input vector is classified as "pass."

$$P(Y=1|x_1,x_2,...,x_{10}) = \frac{1}{1+e^{-(\beta_0+\beta_1x_1+\beta_2x_2+...+\beta_{10}x_{10})}} \quad (2)$$

If *Y* is greater than or equal to 0.5, then the driver is predicted to pass the CDL test; otherwise, the model predicts the driver will fail. Two logistic regression models were created, one with a constant term and one without.

For the initialization of the genetic algorithm, 500 computer programs were generated that predicted a first attempt pass or fail of the CDL test. For fitness evaluation, each of the 500 programs had a fitness value, which was the percentage of the 27 training cases that the program classified correctly. Then the program was asked to classify the test set of 10 cases. The programs that accurately classified at least 80% between the two sets of cases were saved. The programs with the highest fitness value were selected to have the highest probability of continuing to the next generation. Crossover and mutation were used to "reproduce" programs and then they replaced some of the programs from the previous generation. This complete cycle of initialization, evaluation, selection, reproduction, and replacement constitutes one generation; 1000 generations were completed in one run and ten runs were completed.

For comparison of the methods, the values of accuracy, sensitivity, specificity, and validity were examined, based on true positives, true negatives, false positives, and false negatives. The overall model for the logistic regression model (both with and without a constant term) was found to be significant; however, the individual regressors were found not to be significant (alpha = .05). The genetic algorithm was found to become more accurate as the number of generations increased. The genetic algorithm performed better than logistic regression when predicting the outcome of the test cases (those not involved in developing the model). The genetic algorithm was better at sensitivity (true positives) than the logistic regression model, as well as at specificity (true negatives).

This paper provides a model of how logistic regression and genetic algorithms can be used for similar purposes. It provided a comparison of using logistic regression and genetic algorithms to predict the passing or failing of a CDL test. We hope to use genetic algorithms to give an appropriate logistic regression model.

•Knecht (2005)

In his second of two reports, Knecht clarifies and justifies his research results from his first report concerning general aviation pilots' willingness to take off in adverse weather. His concern is having too many independent variables that might cause the model to overfit the data.

Knecht used random number simulation to create a new set of data and use it with the SPSS software package to create a logistic regression model. He concluded that SPSS can still predict takeoffs even with the "noise" data, implying that even with randomly generated data, a model with a good fit can still be created.

He discusses two items of note in his report that are relevant to this research. The first is that the process of stepwise regression. He describes how stepwise regression does not consider all the possible solutions, but instead uses hill-climbing to find a solution from one of the high peaks, which may not be the absolute best solution. The other issue that Knecht discusses concerns the Newton-Rhapson algorithm. The parameter estimation used in the algorithm can explode toward zero or infinity in some cases. SPSS avoids this issue by terminating after a certain number of iterations. However, this produces parameters that are not accurate.

This paper illustrates the use of logistic regression with aviation accident data, supporting our case study. It also justifies an investigation into an alternative method for creating a logistic regression model that does not use the standard practices that have inherent problems.

1.5 **Commentary**

Logistic regression models are created to maximize the likelihood function. However, this procedure is not perfect. Having found that genetic algorithms have been used for logistic regression and that logistic regression has been used with accident data, we seek to create a genetic algorithm that avoids the problems with the maximum likelihood function and that improves the procedure that is commonly used. A case study involving accident data will be used to access our genetic algorithm.

CHAPTER TWO

2 EXPERIENCE AND CRITIQUE OF PASIA

In this chapter, we consider a problem with one independent variable. This may arise, for example, in determining if a patient has a condition, such as diabetes, based on the observation of one real-valued variable. Therefore, our logistic regression model is as follows

$$\mathrm{P}\left(Y=1 \mid x_1\right)=\frac{e^{\left(\beta_0+\beta_1 x_1\right)}}{1+e^{\left(\beta_0+\beta_1 x_1\right)}}=\frac{1}{1+e^{-\left(\beta_0+\beta_1 x_1\right)}} \tag{3}$$

While the applicability of the single independent variable logistic model is arguable, we follow Pasia, *et al.* (2005) in investigating the use of GAs for estimating β_0 and β_1. If GAs cannot compete with other approaches for estimating two parameters, it is doubtful that they will be competitive for estimation of more than two parameters.

We begin this chapter by reporting on the details of previous work by Pasia, *et al*. Based on difficulties in two attempts replicating that work with the basic GA design from Pasia, we then report on the desirable features of a genetic algorithm which forms the base of the rest of this dissertation.

2.1 **Pasia, *et al.*'s Approach**

2.1.1 **Chromosome Representation**

The chromosomes used by Pasia, *et al.* are composed of two arrays, each containing two genes, as shown in Figure 5. The first array represents β_0 and the second

array represents β_1. The first gene of each array is a real number between -1 and 1; the second gene is an integer from the set {10, 20, 30, 40, 50}.

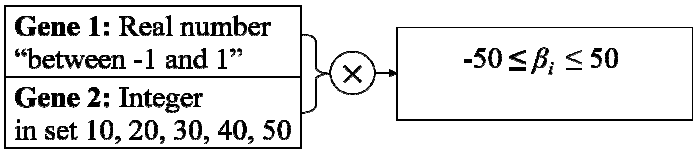

Figure 5: Illustration of Benchmark Chromosome Representation for Each Parameter

2.1.2 **Chromosome Decoding and Evaluation**

Each parameter is estimated with the product of the two genes, as shown in Figure 5. The best solution is the chromosome where the parameters for β_0 and β_1 have the minimal value for $F(\beta_0, \beta_1) = \sum_{i=1}^{60} [Y_i - f(x_i \mid \beta_0, \beta_1)]^2$, since the GA minimizes the least squared error.

2.1.3 **Algorithm**

The initial population (100 chromosomes) is randomly generated. Each member of the new generation is created by modified uniform crossover with tournament. In this crossover, two chromosomes are randomly chosen and the better solution becomes parent 1. Another two chromosomes are randomly chosen, with the better solution becoming parent 2. One offspring will be formed with the first gene of parent 1 and the second

gene of parent 2; the other offspring will contain the first gene of parent 2 and the second gene of parent 1. The offspring with the better solution will move to the next generation. This process is shown with one child being formed in Figure 6.

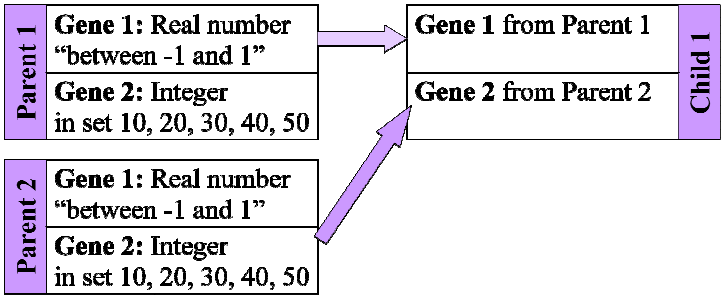

Figure 6: Illustration of Benchmark Crossover for Each Parameter

A chromosome is selected for mutation with 10% probability. A random number between 0.8 and 1.2 is multiplied by the first gene. If this value is greater than one, the first gene of the offspring is then set to one, as shown in Figure 7.

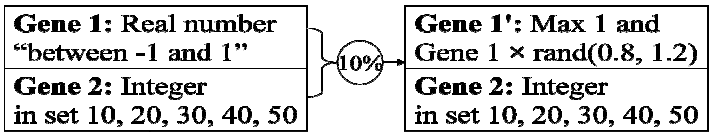

Figure 7: Illustration of Benchmark Mutation for Each Parameter

This new set of chromosomes becomes the next generation. After 100 generations are created, the best chromosome in the last generation is selected as the solution.

2.1.4 **Pasia, *et al.*'s findings**

Pasia, *et al.* report an average β_0 value of 1.1323 and the β_1 average to be -1.675. They therefore conclude that using genetic algorithms for logistic regression is a "viable alternative numerical optimization method." They did not report the value of the error with their GA. (See Table 1 on page 27).

2.1.5 **Our Implementation of Pasia, *et al.*'s Genetic Algorithm**

After several unsuccessful attempts to gain access to the data files used by Pasia, *et al.*, we generate five hundred data sets, each containing 60 observations following their approach closely. The observations (x_i, y_i) of each data set are generated using the values $\beta_0 = 1$ and $\beta_1 = -1$. The sixty values of x_i = -3.0, -2.9, -2.8,…, 2.8, 2.9. The corresponding y_i values for i = 1,…, 60 are shown in Equation (4).

$$y_i = \begin{cases} 1, \text{with probability } f(x_i \mid \beta_0, \beta_1) \\ 0, \text{with probability } 1 - f(x_i \mid \beta_0, \beta_1) \end{cases}, \text{ where } f(x_i \mid \beta_0 = 1, \beta_1 = -1) = \frac{\exp(1 - x_i)}{1 + \exp(1 - x_i)} \qquad (4)$$

We then implement Pasia, *et al.*'s GA design. The average value found for β_0 is -1.164 and the average value found for β_1 is -0.653. The standard deviations (– 2.545 and 1.143, respectively) are higher than those reported. The average least squared error is 29.024 (see Table 1 on page 27).

2.2 **Our Revised Implementation of Pasia,*et al.*'s Genetic Algorithm**

We were unable to replicate Pasia, *et al.*'s findings and we had questions concerning their approach. There was no reason given for their chromosome representation and we could not understand why it was chosen. So we attempted to recreate their genetic algorithm with a different chromosome representation, while retaining the other design elements.

2.2.1 **Chromosome Representation**

We use a real number for our genes, with a binary representation, following Goldberg's original simple GA (Goldberg, 1989). The chromosome representation is composed of two arrays of twenty binary genes. The first twenty binary genes represent β_0, and the second twenty binary genes represent β_1.

The range of values we need to represent is -50 to 50; these endpoints are chosen to correspond to the bounds on the betas set in Pasia, *et al.*'s SAS code. Thus, we need to represent 100 integers. We elect to represent the values to four decimal places. Therefore, in order to determine the number of binary places we need, we calculate $100 \times 10^4 = 1000000$. This tells us that we need $2^n > 1000000$, where n is the number of binary places we need. Solving for n, we see that we need 20 binary places to represent all of the numbers to the accuracy of four decimal places.

2.2.2 **Chromosome Decoding and Evaluation**

Once we have a chromosome of 20 binary digits, we need to "decode" the chromosome into a decimal number. Equation 5 shows the formula that is used.

$$x = Lower + (Base10)\frac{Upper - Lower}{2^{length} - 1} \quad (5)$$

For example, the chromosome 00000100111001011100 would have a base 10 value of 4 + 8 + 16 + 64 + 512 + 1024 + 2048 + 16384 = 20060. The lower and upper values we are representing are -50 and 50, respectively. The length is the number of binary places; for us, that is 20. So our chromosome would decode to the number

$$-50 + (20060)\frac{50 - (-50)}{2^{20} - 1} = -48.0869.$$

Each array is decoded into a real number between -50 and 50 as described above. The best solution is the chromosome where the parameters for β_0 and β_1 have the minimal value for $F(\beta_0, \beta_1) = \sum_{i=1}^{60} [Y_i - f(x_i \mid \beta_0, \beta_1)]^2$, minimizing the least squared error. This is consistent with Pasia *et al.*,'s approach.

2.2.3 **Algorithm**

The overall GA algorithm has the following steps:

1. Initialization – create generation 0, set k=1
2. While stopping criteria not satisfied, form generation *k* from *k*-1:
    a. Perform elite reproduction
    b. Perform single point crossover
    c. Perform mutation, re-evaluate mutated chromosomes
    d. k=k+1
    e. Check stopping criteria

3. Report solution

The following subsections describe the details of these operations.

- Initialization

  The initial population of 100 chromosomes is created by randomly assigning a 0 or a 1 to each gene.

- Forming new generation

  Each new generation is formed through the application of genetic operators. We use elite reproduction for 15%. Elite reproduction is commonly used to ensure the best previous solution is not lost. Therefore, we have included this genetic operator. The other 85% is generated by uniform crossover with tournament selection used to select parents and the surviving child, as described previously. The crossover point is always between β_0 and β_1; a variation would be to allow the crossover point to be at any location in the individual genes.

- Mutation

  Once the new generation of 100 chromosomes is created, we introduce mutation. Each gene of each chromosome (with probability of 0.0001) is toggled – from 0 to 1 or from 1 to 0.

- Stopping Criteria

  A maximum number of generations of 120 is selected for the alternate GA. Pasia, *et al.*'s GA uses 100 generations. The difference is due to our use of

elite reproduction, which reduces the number of new chromosomes evaluated in each generation. In this way, we provide both GAs approximately the same number of chromosome evaluations. After these generations are created, the best chromosome is selected as the solution. Again, the best chromosome is the one with the smallest objective function value.

2.2.4 **Results**

Our alternate GA gives an average β_0 value of -0.566 and an average β_1 value of -0.025. The average OLSE, ordinary least square error, value is 29.732. Using the logistic regression SAS code given in Pasia, *et al.*'s article (see Appendix 1 for the SAS code), we find a β_0 value of -0.47 and a β_1 value of -0.45. The error that SAS reports is 12.34. The procedure Proc logistic yields a β_0 value of -0.50 and a β_1 value of -0.46. The OLSE is not provided. It is difficult to conclude that Pasia, *et al.*'s approach or our approach is a good alternative, given the results produced by SAS.

While GA research often considers running time to be an interesting factor, we do not consider it because the computational environment in which we run our experiments is not the same as the environment that would be used when running SAS. Moreover, this dissertation provides a proof of concept, not a market-ready implementation.

Table 1: Summary of Computational Experiments

	β_0 **(intended to be 1)**			β_1 **(intended to be -1)**			LSE		
	Mean	Median	Std. Dev.	Mean	Median	Std. Dev.	Mean	Median	Std. Dev.
Pasia	1.132	1.011	.550	-1.675	-0.980	0.922	*	*	*
Replication of Pasia	-1.164	-0.764	2.545	-0.653	-0.705	1.143	29.025	28.869	0.720
Our Alternate	0.566	0.808	2.890	0.025	-0.255	3.323	29.732	29.669	0.808
proc nlin from SAS	-0.466	-0.446	0.287	-0.454	-0.438	0.183	12.336	12.457	1.197
proc logistic from SAS	-0.497	-0.478	0.285	-0.465	-0.452	0.180	*	*	*

* Not reported

2.3 **Critique of Pasia, *et al.***

We attempted to replicate the approach of Pasia, *et al.*, but did not achieve satisfactory results. We also tried to replicate their environment, but with a different representation, also with poor results. Therefore, we began to question their overall approach and determined the following shortcomings which our work inherently continued.

- Chromosome Representation

Pasia, *et al*. use the limits of -50, +50 for the chromosomes. The choice for the representation using two genes to get a value between -50 and +50 is not justified, when real-valued representations have been known in the literature for 20 years. We instead prefer the genetic algorithm to be able to take on any values possible based on the limits of the software.

- The Objective Function

In linear regression, there are several basic assumptions. One of these assumptions is that the variance of the error variable is constant, a condition known as homoscedasticity. When the variance of the residuals depends on the independent variables, this assumption fails to hold true. When this happens, the estimates for the regression coefficients will no longer have the smallest variance. They still are unbiased, just no longer efficient. When homoscedasticity is not present, tests for statistical significance become inaccurate (Menard, 2002).

When all of the assumptions hold, the least squares estimators (sometimes referred to as OLS estimators – ordinary least squares estimators) possess two very important properties. The first is that the estimators are unbiased (the mean of the estimate of a parameter over an infinite number of random samples will be equal to the parameter it is estimating). The second is that the estimators are efficient,

meaning they have the smallest variance of a set of unbiased estimators (Berry, 1993).

Since we have an indicator response variable, the error variable does not have a constant variance. Therefore, a method other than the ordinary least squares must be utilized to estimate our coefficients. Maximum likelihood estimation is also used frequently with logistic regression due to the lack of homoscedasticity inherent in a logistic regression model. With this method, the estimates of the coefficients are chosen in such a manner as to give the best chance of seeing the observed outcomes of the set of data. Hence, they are the estimates that are "most likely" to cause observations that are in our data sample. The likelihood function for logistic regression is $LF = \prod \left[P_i^{Y_i} * (1 - P_i)^{1-Y_i} \right]$, where P_i is the probability that the function value is one. Sometimes the log likelihood function is maximized in order to avoid dealing with very small numbers that will occur when multiplying many probabilities. Its form is $\ln LF = \sum \left\{ \left[Y_i * \ln P_i \right] + \left[(1 - Y_i) * \ln(1 - P_i) \right] \right\}$ (Pampel, 2000).

In our third version of a GA, we have used the log likelihood function for our objective function. In order to maximize this value, those chromosomes with the higher objective function value are moved to the next generation.

- Independent Variables

The algorithm developed by Pasia, *et al.* only allows for one independent variable. In many applications, more than one independent variable is needed. We would like our genetic algorithm to be generic enough so that the independent variables are not specified when creating the model. Limiting the GA to one independent variable is not practical or general enough for common use.

- Constant

The other potential problem is that the genetic algorithm of Pasia, *et al.* always includes a constant in the logistic regression model. Again, in other applications it may not be the case that a constant needs to be included in the model. We would eventually like our GA to determine whether the constant is included or not.

- Validation Method

Pasia, *et al*. validate their GA by comparing their results to the procedure "proc nlin" in SAS. This does not seem to be comparing two similar items. The validation should be made by comparing the results with the "proc logistic" procedure in SAS or the implementation of the "proc logistic" methodology itself.

Therefore, this dissertation addresses these shortcomings as described in the following.

- Consideration of alternate objective function

Pasia, *et al.* uses the minimization of least squares as the objective function and results from *proc nlin* in SAS to validate the results. However, *proc logistic* in SAS utilizes the maximum likelihood function for the objective, leading us to also utilize the maximum likelihood function for our new objective function. We utilize the natural log as shown in Equation 6.

$$\ln lf = \sum_i y_i \ln\left(\frac{1}{1+e^{\left(-\beta_0-\sum_k \beta_k x_k\right)}}\right) + \sum_i (1-y_i)\ln\left(1-\frac{1}{1+e^{\left(-\beta_0-\sum_k \beta_k x_k\right)}}\right) \qquad (6)$$

The SAS software implementation of *proc logistic* uses the Iteratively Reweighted Least Squares (IRLS) method to compute the objective function. In order to validate our work, we implement this method in our code, as opposed to using SAS directly; while SAS is readily available to us, it is only the computational results that SAS produces that are interest to us, not the actual software environment. The method used to compute the Iteratively Reweighted Least Squares is based on the Newton-Rhapson method. See Appendix B for the details of this procedure.

There is an issue with computing ability to represent very large or very small numbers, due to the use of double precision data types. While infinite precision data types exist, and in fact, SAS and SPSS are implemented with those data types, mastery of those skills is beyond the scope of this dissertation.

- Chromosome representation

Instead of the approach of Pasia, *et al.* to represent each parameter as a value between -50 and 50, utilizing the multiplication of two genes, or a binary representation that has to be decoded into a real number, we decided to represent each parameter (β_0, β_1, etc. as needed) as real numbers. One advantage of this is that the limits on the parameters are just the limits imposed by the computing environment, as opposed to a pre-defined limit of -50 to 50.

2.4 **Post-Exploratory Resolutions**

Since one of our main concerns is the limitation of only one independent variable, we create a data set (see Appendix C) with multiple independent variables. We continue our investigation with this data set instead of proceeding with the data from Pasia, *et al.*

We also investigate the procedure that software packages, such as SAS, implement to find the logistic regression model. We find that SAS seeds its answer with the Ordinary Least Squares Error (OLSE). In order to give our genetic algorithm a fair chance to compete with a software package, we evaluate different types of seeding of our genetic algorithm with the OLSE and the IRLS answers in the initial population. Code is written to mathematically replicate the procedure that SAS uses.

Through our examination of the work of Pasia, *et al.*, we find several flaws which are presented here. The following chapters will demonstrate how features to overcome these flaws have been incorporated into the genetic algorithm for logistic regression.

CHAPTER THREE

3 GA TO DECIDE CONSTANT INCLUSION

In the previous chapter, we explore the use of Pasia, *et al.*'s GA as a viable starting point for the design of a GA for logistic regression. After extensive experimentation, we discover several flaws in their GA design. In this chapter, we focus on designing a GA that does not require the user to decide *a priori* whether the model should include the constant term or not, but instead incorporates that decision. We simultaneously present a chromosome design that allows for more than one independent variable.

3.1 **Chromosome Representation and Required GA Changes**

The chromosome consists of genes representing the coefficients for the independent variables in the given data set and a gene that represents whether or not to include the constant term in the logistic regression model. Figure 8 illustrates this chromosome representation.

β_0	β_1	β_2	β_3	…….	β_k	Constant: Yes / No

Figure 8: Chromosome Representation

Consider the chromosome β_0 = 34.3203, β_1 = -1.1344, β_2 = 10.9752, constant = yes. The logistic regression model to be considered is shown in equation (7).

$$y(\beta_0, \beta_1, \beta_2) = \frac{1}{1+e^{-(34.3203-1.1344x_1+10.9752x_2)}} \tag{7}$$

Consider in contrast, the same chromosome, but constant = no. The logistic regression model now becomes that of equation (8).

$$y = \frac{1}{1+e^{-(-1.1344x_1+10.9752x_2)}} \tag{8}$$

The necessary changes to previous GAs are (1) introducing a range in which the initial gene values must lie and (2) changing the mutation genetic operator. The upper value of the range that is used to initialize the genes for the coefficients of the independent variables is generated by taking the largest coefficient that is generated by the OLSE solution and the IRLS solution. That value is then rounded up to the next integer value to give the upper bound for the range. The lower value of the range uses the smallest coefficient generated by those two solutions. That smallest value then gets rounded down to the previous integer to give the lower bound for the range. We initialize our genes within this range in order to limit the search of our genetic algorithm. With small coefficients, allowing the GA to search the entire solution space is not reasonable, so we begin the search within a certain area.

We also need to consider a different mutation for the genes, which has two components, reflecting that the genes have two components as well. For the genes that represent the coefficients that get mutated, we randomly generate a real number between the upper coefficient mentioned earlier in the appropriate range for initialization and the

lower coefficient. This value gets added to the original coefficient. For the inclusion genes, mutation simply acts as a toggle between "Yes" and "No". So if the constant gene is chosen to undergo mutation, the original gene is switched to the alternative (Yes to No or No to Yes).

The overall algorithm we implement in this chapter is as follows:

1. Initialization – create generation 0, set k=1
2. While stopping criteria not satisfied, form generation *k* from *k*-1:
    a. Perform elite reproduction
    b. Perform crossover
    c. Perform mutation, re-evaluate mutated chromosomes
    d. k=k+1
    e. Check stopping criteria
3. Report solution

3.2 **Computational Experiments**

Several crossover genetic operators, such as single point crossover (SPC), two point crossover (TPC), and parametric uniform crossover (PUC), are considered. Three genetic algorithms (one for each type of crossover) are subjected to 30 replications with our example data, shown in Appendix C.

3.2.1 **Unseeded Initial Population**

In order to see the outcome from a pure genetic algorithm, initially we do not seed the initial population with the results from the OLSE or the IRLS method. The results of these tests are given in Table 2. The values given are the objective function values for

the following cases: constant must be included, constant is not included, GA determines the constant inclusion.

Table 2: Results without GA being seeded

		SPC	TPC	PUC
with constant	Max	-12.30594	-12.30591	-12.30592
	Avg	-12.30623	-12.30604	-12.30606
	Std Dev	0.00028	0.00015	0.00015
without constant	Max	-12.32222	-12.32220	-12.32217
	Avg	-12.32265	-12.32228	-12.32233
	Std Dev	0.00045	6.83471E-05	0.00013
constant decided by GA	Max	-12.30594	-12.30590	-12.30592
	Avg	-12.30633	-12.30609	-12.30602
	Std Dev	0.00044	0.00018	9.51485E-05

The IRLS method yields a maximum likelihood objective value of -12.30589 for the model with a constant and -12.32217 for the model without a constant. Notice for all three crossover techniques, the maximum value of the objective function of the 30 runs is the same to within four decimals and the average is the same to within three decimal places. The standard deviations for each model are quite small (< 0.0005), indicating an adequate stopping criterion was used; we used 50,000 generations. Depending on the user's precision preference, seeding may not be necessary.

3.2.2 **Seeded Initial Population**

The data was subjected to 30 runs again; this time the initial population was seeded with the chromosomes for the OLSE and the IRLS methods; the remaining 98 chromosomes in the initial population were generated randomly. In this case, the results when the constant is determined by the GA should be no worse than the better of the results of the GAs with either the constant required or forbidden. The results are given in Table 3.

Table 3: Results with seeded GA

		SPC	**TPC**	**PUC**
with constant	Max	-12.305886	-12.305886	-12.305886
	Avg	-12.305886	-12.305886	-12.305886
	Std Dev	5.31279E-07	5.31279E-07	5.31279E-07
without constant	Max	-12.322165	-12.322165	-12.322165
	Avg	-12.322165	-12.322165	-12.322165
	Std Dev	1.77093E-07	1.77093E-07	1.77093E-07
constant decided by GA	Max	-12.305886	-12.305886	-12.305886
	Avg	-12.305886	-12.305886	-12.305886
	Std Dev	5.31279E-07	5.31279E-07	5.31279E-07

With the seeded genetic algorithm, the maximum values as well as the average values of the objective function for all three crossover techniques match the objective

function that is produced by the IRLS algorithm, used in SAS. The standard deviations have been decreased from the previous non-seeded GA.

3.3 **Discussion**

In our different populations, we find two or more chromosomes with the same objective function value but different beta coefficient values. In other words, our GA finds several different models that evaluate to the same fitness. Small differences in the beta values do not create differences in the objective function. Secondary statistical measures could be used to prioritize these models, if a modeler was interested in distinguishing between several initially equivalent models.

We find two point crossover and parametric uniform crossover to be comparable. TPC had higher objective function values and lower standard deviations about half of the time and SPC did the other half of the time. Single point crossover never dominates the other two crossover methods for any measurement.

3.4 **Conclusions**

In this chapter, we develop and evaluate the ability of a GA to determine whether or not a constant term should be included in a logistic regression model. Based on comparisons to IRLS generated logistic regression models, we conclude that a GA designed as proposed can in fact reasonably determine whether a model with or without a constant provides a better maximum likelihood value for a given set of data. These results allow us to conclude that genetic algorithms can be competitive with the available software for creating logistic regression models.

CHAPTER FOUR

4 GA TO DECIDE INDEPENDENT VARIABLE INCLUSION

It is clear that a modeler may not want to include all of the independent variables given in a data set. In this case, the chromosome needs to allow for the relevant independent variables to be included or not.

4.1 **Chromosome Representation and Required GA Changes**

To address this concern, the chromosome is adapted, following the model from the previous section. We add genes that represent the decision to include a particular independent variable in the logistic regression model. A representation of a chromosome is given below in Figure 9.

β_0	β_0 incl	β_1	β_1 incl	β_2	β_2 incl	β_3	β_3 incl	…….	β_k	β_k incl

Figure 9: Chromosome Representation

For example, given the chromosome β_0= 4.9027, $\beta_{0_{\text{incl}}}$ = 1, β_1= 4.3546, $\beta_{1_{\text{incl}}}$ = 1, β_2= -0.9901, $\beta_{2_{\text{incl}}}$ = 0, β_3= -2.6774, $\beta_{3_{\text{incl}}}$ = 1 will have the corresponding logistic regression model in equation (9).

$$y = \frac{1}{1+e^{-(4.9027+4.3546x_1-2.6774x_3)}} \tag{9}$$

The genetic operator of mutation is used in an equivalent fashion as in the previous chapter.

The overall algorithm we implement in this chapter is as follows:

1. Initialization – create generation 0, set k=1
2. While stopping criteria not satisfied, form generation *k* from *k*-1:
    a. Perform elite reproduction
    b. Perform crossover
    c. Perform mutation, re-evaluate mutated chromosomes
    d. k=k+1
    e. Check stopping criteria
3. Report solution

4.2 **Computational Experiments**

Following the methodology in the previous chapter, several crossover genetic operators are considered: single point crossover, two point crossover, and parametric uniform crossover. Three genetic algorithms (one for each type of crossover) are subjected to 30 replications with our example data. Again, 50,000 generations are used as the stopping criteria. The GA is run both with and without seeding the initial population. The results of these tests are given in Table 4.

Table 4: GA results

		SPC	TPC	PUC
not seeded	Max	-12.3059	-12.3059	-12.3059
	Avg	-12.3063	-12.3066	-12.3060
	Std Dev	0.0003	0.0030	7.9080E-05
seeded	Max	-12.3059	-12.3059	-12.3059
	Avg	-12.3059	-12.3059	-12.3056
	Std Dev	5.3128E-07	5.3128E-07	5.3128E-07

Recall that the IRLS method yields a maximum likelihood objective value of -12.305886 for the model with a constant and -12.322165 for the model without a constant. So our GA finds the better model of including the constant both with and without seeding. The maximum values of our objective function over the thirty runs are equivalent to IRLS procedure to four decimal places and the average values are the same to three decimal places.

4.3 **Discussion**

Again, we see that using a genetic algorithm to generate logistic regression models is competitive to current software packages which implement IRLS. Using genetic algorithms instead of IRLS avoids the issues of parameter estimates sometimes going toward zero or infinity with the Newton-Rhapson method that has mentioned by other researchers (Knecht 2005). Additionally, we see that using a genetic algorithm can remove any user bias by having the GA make the decisions about the independent

variables and constant instead of the user. This is where the use of genetic algorithms differs from current available software packages. A modeler would typically be asked to enter each independent variable. The order in which he would do so might be influenced by the modeler's preconceived notions of importance. Also, given a data set with twenty independent variables, one modeler may enter the first eighteen thinking only those are important in predicting the outcome. Yet another modeler using the same data set may enter only the first fifteen, viewing only those as important. Using a genetic algorithm removes this chance of user bias. The independent variables are all equal in the eyes of the GA. It is only when the best logistic regression model is found that we discover which independent variables should be included in the model.

4.4 **Conclusions**

In this chapter, we develop and evaluate the ability of a GA to determine whether or not any individual term, for the constant or the independent variables, should be included in a logistic regression model. Based on comparisons to IRLS generated logistic regression models, we conclude that a GA designed as proposed can in fact be competitive with the available software for creating logistic regression models.

CHAPTER FIVE

5 Case Study

This case study demonstrates the viability of the proposed methodology for the development of logistic regression models using genetic algorithms. Data for the case study have been provided by Dr. Scott A. Shappell, Professor in the Industrial Engineering department at Clemson University. The data describes characteristics of general aviation accidents. The outcome (fatal or non-fatal) of the accident is the binary response variable. The independent variables will consist of several different data types. We have 113 initial independent variables, some of which are categorical, some are binary, and some are continuous. We will subject this data to the methodology developed in the previous chapter.

5.1 **Data**

Some of the original set of 113 independent variables have missing data due to different states reporting different items in the accident report. Other variables are not of interest when trying to predict whether a crash will be fatal or not, such as year. With the aid of Dr. Shappell, we have extracted thirteen independent variables and created a data set that has five hundred observations, which is representative of factors that might influence the outcome of an aviation accident.

Ten of the independent variables are binary, nine of which relate to human factors. These independent variables include:

- Weather (0 = no weather conditions present 1 = weather)

- DE – decision errors (0 = no 1 = yes)
- SBE – skill-based errors (0 = no 1 = yes)
- PE – perception errors (0 = no 1 = yes)
- V – violations (0 = no 1 = yes)
- AMS – adverse metal state (0 = no 1 = yes)
- APS – adverse psychological states (0 = no 1 = yes)
- PML – physical/mental limitations (0 = no 1 = yes)
- CRM – crew resource management (0 = no 1 = yes)
- IS – inadequate supervision (0 = no 1 = yes)

See Appendix D for the data.

5.2 Chromosome Representation

We use the chromosome representation that was presented in Chapter 4 (see Figure 9). There is a gene for each independent variable that uses a real number to represent the coefficient and a gene for each independent variable that uses a 0 or 1 to represent whether to include the coefficient in the model (1) or not (0).

5.3 Chromosome Decoding and Evaluation

The chromosome decodes into a logistic regression model in which each coefficient is the raw coefficient multiplied by the include value. The objective function that is being maximized is the natural log of the likelihood function (see Equation 3).

5.4 **Algorithm**

The overall GA algorithm has the following steps:

1. Initialization – create generation 0, set k=1
2. While stopping criteria not satisfied, form generation *k* from *k*-1:
    a. Perform elite reproduction
    b. Perform two point crossover
    c. Perform mutation, re-evaluate mutated chromosomes
    d. k=k+1
    e. Check stopping criteria
3. Report solution

The following subsections describe the details of these operations.

- Initialization

The initial population of 100 chromosomes is created by randomly assigning a real number within an appropriate range to each gene that represents a coefficient and a 0 or a 1 to each gene that represents inclusion. The range is generated in the same manner as was described in Chapter 4. We did not seed the initial population.

- Forming new generation

Each new generation is formed through the application of genetic operators. We use elite reproduction for 15%. The other 85% is generated by two point crossover with tournament selection used to select parents and the surviving child, as described

previously. The crossover points are randomly selected. Once the new generation of 100 chromosomes is created, we introduce mutation.

- Mutation

Each gene of each chromosome (with probability of 0.0001) is mutated. A gene representing a coefficient is mutated in the manner described in Chapter 4 and a gene representing inclusion is toggled – from 0 to 1 or from 1 to 0.

- Stopping Criteria

After 100,000 generations are created, the best chromosome is selected as the solution. The best chromosome is that with the largest objective function. We also run the GA using 200,000 and 500,000 generations.

5.5 **Results**

The resulting estimated parameters from applying the IRLS procedure to the data are shown in Table 5, along with the corresponding likelihood value of -303.964. Table 5 also gives our results in terms of the parameters for the best model in each of our stopping criteria. Our results for the objective function are included. Although our results do not yield an objective function that is as large as the two software packages, we feel we have a comparable outcome. The maximum and average of the objective function over the thirty runs increases as the number of generations increases, while the standard deviation decreases.

Table 5: Case study parameter results

	Maximum number of generations			
parameter	100000	200000	500000	IRLS
$\beta_0(const)$	1.521	1.520	1.535	1.998
$\beta_1(month)$	0.012	0.018	0.020	0.022
$\beta_2(time)$	0	0	0	-0.0002
$\beta_3(injury\_total)$	0	0	0	-0.0997
$\beta_4(weather)$	-0.995	-0.972	-0.986	-1.118
$\beta_5(DE)$	-0.270	-0.281	-0.289	-0.318
$\beta_6(SBE)$	0	-0.076	-0.092	-0.090
$\beta_7(PE)$	-0.504	-0.510	-0.513	-0.505
$\beta_8(V)$	-0.1740	-0.182	-0.192	-0.164
$\beta_9(AMS)$	0.155	0.162	0.158	0.195
$\beta_{10}(APS)$	-0.124	-0.094	-0.097	-0.037
$\beta_{11}(PML)$	0.606	0.612	0.611	0.616
$\beta_{12}(CRM)$	-0.349	-0.369	-0.380	-0.340
$\beta_{13}(IS)$	0.085	0.106	0.110	0.120
avgLn log value	-304.403	-304.575	-304.521	-303.964
std	0.161	.150	0.148	------
max	-304.403	-304.397	-304.396	------

In our case study our GA generates two models with the same objective function value of -304.442. However, the two models are very different in that each includes different independent variables. Model 1 does not include the APS term and model 2 does not include the IS term. Table 6 gives the two different models' parameter estimates. This might allow for flexibility among users and when problems with data occur.

Table 6: Comparison of models having the same objective function value

Objective Function Value = -304.442		
parameter	Model 1	Model 2
$\beta_0(const)$	1.611	1.626
$\beta_1(month)$	0.027	0.020
$\beta_2(time)$	0	0
$\beta_3(injury\_total)$	0.098	-0.088
$\beta_4(weather)$	-1.090	-1.028
$\beta_5(DE)$	-0.270	-0.292
$\beta_6(SBE)$	-0.047	-0.067
$\beta_7(PE)$	-0.497	-0.516
$\beta_8(V)$	-0.144	-0.142
$\beta_9(AMS)$	0.187	0.192
$\beta_{10}(APS)$	**0**	**-0.046**
$\beta_{11}(PML)$	0.605	0.606
$\beta_{12}(CRM)$	-0.341	-0.366
$\beta_{13}(IS)$	**0.121**	**0**

5.6 **Conclusions**

This case study demonstrates that genetic algorithms can create viable models for logistic regression using different types on input data. It has been validated on real-world data.

6 CONCLUSIONS AND AREAS FOR FUTURE RESEARCH

6.1 **Conclusions**

This dissertation begins by attempting to replicate a benchmark genetic algorithm by Pasia, *et al.* to develop a logistic regression model. Several problems with their approach are uncovered through this process. The chromosome representation and the objective function are the two major issues that need to be resolved. This dissertation reports on the development of a genetic algorithm with a more natural chromosome representation and a more appropriate objective function for use with logistic regression models.

Further investigation leads to a more general genetic algorithm for logistic regression through an iterative process. The proposed genetic algorithm has the following novel features:

- The GA can create a model with an arbitrary number of independent variables.
- The GA can determine whether or not a particular data set should be modeled with logistic regression model that includes the constant term.
- The GA can determine which independent variables should be included in a logistic regression model for a particular data set and estimate the parameters.

This proposed genetic algorithm is used in a case study of general aviation data. We find results that are comparable to those given by current software packages. The added value of using genetic algorithms for logistic regression is that of reducing user bias. By using genetic algorithms, the modeler does not have the opportunity to include

his or her preferences into the model building process. The genetic algorithm completely decides which independent variables should be included in the model.

6.2 **Applications**

While our research uses genetic algorithms to generate a logistic regression model for a set of data concerning aviation, their use is applicable in many areas. GAs could be used to predict the success of a hotel or restaurant at a certain location, predict accidents in a factory, or predict the probability of mortality in a certain medical scenario. The useful characteristic is that user bias is removed to generate the logistic regression model. Different people could use the same data and generate the same model. This is probably not common, given that with current software each modeler chooses which variables to include and in which order to include them. Genetic algorithms also allow for the possibility of having two different models, both of which are equally valid. In this way, a modeler is allowed some flexibility without losing significant accuracy of the solution compared to iterative methods such as IRLS.

6.3 **Future research**

There are several areas in which further work or research is warranted. The first of these concerns the issue of data precision. We used the computer language of C to write our genetic algorithm. C uses double precision which does not allow for very large or very small numbers to be expressed. In some of our calculations, we have to create a bound that will stop the program from crashing due to the inability to store these very small numbers. Using a language that allows for infinite precision would remove some

of the problems we encounter with our calculations with matrices. With this improved accuracy we would hope to see improved results in terms of the data sets we could handle.

Another aspect that would benefit from more research is to make our genetic algorithm more user-friendly. Our current genetic algorithm is able to generate the best model according to the maximum likelihood objective and report it to the user. However, we see that sometimes there are several models with the same objective function value. In practice, there could be times when there are several "best" models. We would like the genetic algorithm to be able to allow the user to choose between those models that are considered "best"; this would allow for flexibility when using genetic algorithms for logistic regression. This allows the decision-maker to consider several models, introducing the user's knowledge in the problem domain after model creation, as opposed to before model creation. In this way, the user can be exposed to models that previously would have been eliminated due to their bias in model creation.

Considering alternative measures for the objective function is still another path for continuing our investigation. The GA of Pasia, *et al.* minimized least squared error as the objective. The proposed GA maximizes the likelihood function, consistent with the IRLS procedure. While this matches the procedures implemented in software such as SAS and SPSS, users may use other statistical measures when selecting between different models. By including these measures, perhaps in a multi-objective sense, we may be able to provide users with several more interesting models from which to select.

Our investigation of genetic algorithms for the use of logistic regression has shown comparable results to current practice. It may be however, that other metaheuristics such as simulated annealing or tabu search may be better suited to logistic regression. With the favorable results from genetic algorithms we achieve, we believe this is a strong effort toward institutionalizing the use of metaheuristics in logistic regression.

APPENDICES

Appendix A

SAS code

The following is the SAS code provided by Pasia.

```
proc nlin data = <insert data-set name here> maxiter = 32000 method =
newton;
parms beta0=0 beta1=0; bounds beta0 beta1 <50;
bounds beta0 beta1 >-50;
A = exp(beta0 + beta1 * x);
pie = A/(1+A);
model y= pie;
der = A/(1+A)**2;
der.beta0 = der;
der.beta1 = der*x;
run;

proc logistic data = <insert data-set name here> descending;
model y= x;
run;
```

Appendix B

Iteratively Reweighted Least Squares Procedure

This appendix is adapted from Montgomery, Peck, and Vining (2001) for convenience of the reader.

In a logistic regression model, the log-likelihood function is

$$\ln L(y,\beta) = \sum_{i=1}^{n} y_i x_i'\beta - \sum_{i=1}^{n} \ln\left[1+\exp(x_i'\beta)\right]$$

Oftentimes in logistic regression models, there are repeated observations for the levels of the x variables. If we let y_i be the number of 1's observed for the i^{th} observation and n_i be the number of trials at each observation, then the log-likelihood function can be written as

$$\ln L(y,\beta) = \sum_{i=1}^{n} y_i\pi_i + \sum_{i=1}^{n} n_i \ln(1-\pi_i) - \sum_{i=1}^{n} y_i \ln(1-\pi_i)$$

An iteratively reweighted least-squares algorithm is used to calculate the maximum likelihood estimates. The solutions to $\frac{\partial L}{\partial \beta} = 0$ are the MLEs. This equation can be written as $\frac{\partial L}{\partial \pi_i}\frac{\partial \pi_i}{\partial \beta} = 0$

Now

$$\frac{\partial L}{\partial \pi_i} = \sum_{i=1}^{n} \frac{n_i}{\pi_i} - \sum_{i=1}^{n} \frac{n_i}{1-\pi_i} + \sum_{i=1}^{n} \frac{y_i}{1-\pi_i}$$

And

$$\frac{\partial \pi_i}{\partial \beta} = \left\{ \frac{\exp(x_i'\beta)}{1+\exp(x_i'\beta)} - \left[\frac{\exp(x_i'\beta)}{1+\exp(x_i'\beta)} \right]^2 \right\} x_i$$

Combining these, we get

$$\frac{\partial L}{\partial \beta} = \left[\sum_{i=1}^{n} \frac{n_i}{\pi_i} - \sum_{i=1}^{n} \frac{n_i}{1-\pi_i} + \sum_{i=1}^{n} \frac{y_i}{1-\pi_i} \right] \pi_i (1-\pi_i) x_i$$

$$= \sum_{i=1}^{n} \left[\frac{y_i}{\pi_i} - \frac{n_i}{1-\pi_i} + \frac{y_i}{1-\pi_i} \right] \pi_i (1-\pi_i) x_i$$

$$= \sum_{i=1}^{n} (y_i - n_i \pi_i) x_i$$

So the maximum likelihood estimator solves

$$X'(y-\mu) = 0$$

Where $y' = \left[y_1, y_{2,}, ..., y_n \right]$ and $\mu' = \left[n_1\pi_1, n_2\pi_2, ..., n_n\pi_n \right]$. "This set of equations" is known as the maximum likelihood score equations. With the linear regression model $E(y) = X\beta = \mu$, the normal equations are $X'X\hat{\beta} = X'y$. This can be written as $X'(y - X\beta) = 0$ or $X'(y-\mu) = 0$, taking the same form as the maximum likelihood score equations.

To solve these score equations, the Newton-Rhapson method is used. Using a first-order Taylor series expansion we approximate

$$p_i - \pi_i \approx \left(\frac{\partial \pi_i}{\partial \beta} \right)' \left(\beta^* - \beta \right) \tag{1}$$

where

$$p_i = \frac{y_i}{n_i}$$

and β^* is the value of β that solves the score equations. Letting

$$\eta_i = x_i' \beta$$

we have

$$\frac{\partial \eta_i}{\partial \beta} = x_i$$

We can see that

$$\pi_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}$$

Using the chain rule we get

$$\frac{\partial \pi_i}{\partial \beta} = \frac{\partial \pi_i}{\partial \eta_i} \frac{\partial \eta_i}{\partial \beta} = \frac{\partial \pi_i}{\partial \eta_i} x_i$$

So we can rewrite equation (1) as the following

$$p_i - \pi_i \approx \left(\frac{\partial \pi_i}{\partial \eta_i} \right) x_i' (\beta^* - \beta)$$

$$p_i - \pi_i \approx \left(\frac{\partial \pi_i}{\partial \eta_i}\right)(x_i'\beta^* - x_i'\beta)$$

$$p_i - \pi_i \approx \left(\frac{\partial \pi_i}{\partial \eta_i}\right)(\eta_i^* - \eta_i) \qquad (2)$$

where η_i^* is the value of η_i evaluated at β^*. Notice that

$$(y_i - n_i\pi_i) = (n_i p_i - n_i\pi_i) = n_i(p_i - \pi_i)$$

and

$$\pi_i = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}$$

Therefore, we can write

$$\frac{\partial \pi_i}{\partial \eta_i} = \frac{\exp(\eta_i)}{1+\exp(\eta_i)} - \left[\frac{\exp(\eta_i)}{1+\exp(\eta_i)}\right]^2 = \pi_i - \pi_i^2$$

So now we have

$$y_i - n_i\pi_i \approx \left[n_i\pi_i(1-\pi_i)\right](\eta_i^* - \eta_i)$$

A first approximation to the variance of the linear predictor $\eta_i^* = x_i'\beta^*$ is

$$Var(\eta_i^*) \approx \frac{1}{n_i\pi_i(1-\pi_i)}$$

So,

$$y_i - n_i\pi_i \approx \left[\frac{1}{Var(\eta_i^*)}\right](\eta_i^* - \eta_i)$$

And the score equations can be written as

$$\sum_{i=1}^{n}\left[\frac{1}{Var(\eta_i)}\right](\eta_i^* - \eta_i) = 0$$

In matrix notation, this becomes $X'V^{-1}(\eta^* - \eta) = 0$ where V is a diagonal matrix of the weights formed from the variances of η_i. Since $\eta = X\beta$ the score equations can be written as $X'V^{-1}(\eta^* - X\beta) = 0$ and the maximum likelihood estimate of β is

$$\hat{\beta} = (X'V^{-1}X)^{-1}X'V^{-1}\eta^*$$

The problem is that we do not know η^*. However, we can use Equation (2) :

$$p_i - \pi_i \approx \left(\frac{\partial \pi_i}{\partial \eta_i}\right)(\eta_i^* - \eta_i)$$

Solving for η_i^*, we get

$$\eta_i^* \approx \eta_i + (p_i - \pi_i)\frac{\partial \eta_i}{\partial \pi_i}$$

Let $z_i = \eta_i + (p_i - \pi_i)(\partial \eta_i)/(\partial \pi_i)$ and $z' = [z_1, z_2, ..., z_n]$. Then the Newton-Rhapson estimate of β is

$$\hat{\beta} = (X'V^{-1}X)^{-1}X'V^{-1}z$$

The random part of z_i is

$$(p_i - \pi_i)\frac{\partial \eta_i}{\partial \pi_i}$$

So,

$$Var\left[(p_i-\pi_i)\frac{\partial\eta_i}{\partial\pi_i}\right]=\left[\frac{\pi_i(1-\pi_i)}{n_i}\right]\left(\frac{\partial\eta_i}{\partial\pi_i}\right)^2$$

$$=\left[\frac{\pi_i(1-\pi_i)}{n_i}\right]\left(\frac{1}{\pi_i(1-\pi_i)}\right)^2$$

$$=\frac{1}{n_i\pi_i(1-\pi_i)}$$

So, V is the diagonal matrix of weights formed from the variances of the random part of z. The IRLS algorithm based on the Newton-Rhapson method can be summarized as:

1. Use ordinary least squares to obtain an initial estimate of β, say $\hat{\beta}_0$
2. Use $\hat{\beta}_0$ to estimate V and π
3. Let $\eta_0 = X\hat{\beta}_0$
4. Base z_1 on η_0
5. Obtain a new estimate $\hat{\beta}_1$, and iterate until some suitable convergence criterion is satisfied.

Appendix C

Example Data

y	x1	x2	x3	x4	x5
1	3	12	6	20	8
0	3	10	3	18	9
0	5	11	7	20	7
0	7	9	7	15	10
1	5	11	5	19	8
1	3	10	3	16	6
0	6	7	7	18	5
0	7	9	7	15	8
1	4	8	2	13	3
1	8	13	8	14	9
1	9	18	9	23	12
0	6	8	6	17	8
0	5	11	5	19	6
1	2	7	2	10	4
1	2	8	2	9	3
1	5	11	5	19	7
0	3	10	4	12	8
0	4	7	8	15	9
0	5	15	8	14	10
0	9	18	9	23	12
1	7	11	6	15	9
1	6	12	8	14	9
1	5	9	4	11	6
0	4	8	6	15	7
0	1	5	2	8	3

Appendix D

Case Study Data

Fatal	Month	Time	Injury_Total	Weather	DE	SBE	PE	V	AMS	APS	PML	CRM	IS
1	8	1216	2	0	1	1	0	1	0	0	0	0	0
1	9	2123	1	0	0	1	0	1	0	0	1	0	0
1	11	1330	1	0	0	1	0	1	0	0	0	0	0
1	6	1305	2	0	1	0	0	0	0	0	0	0	0
0	5	2222	1	0	0	1	0	1	0	0	0	0	0
0	8	2200	7	0	0	0	0	1	1	0	0	0	0
1	11	1650	1	0	0	0	0	1	0	1	0	0	0
1	11	2319	2	0	1	0	0	1	1	0	0	0	0
0	12	1530	2	0	0	1	0	1	0	0	1	0	0
1	4	1130	6	0	0	1	0	1	1	0	1	0	0
1	9	1514	2	0	0	1	0	0	0	0	0	0	0
1	1	335	2	0	1	0	0	1	0	0	0	0	0
0	11	1805	5	0	0	1	0	1	0	0	0	0	0
1	2	1140	1	0	1	1	0	0	0	1	0	0	0
0	3	1450	1	0	0	1	0	1	1	0	0	0	0
0	3	2255	4	0	1	0	0	1	0	0	0	0	0
1	3	648	1	0	0	0	0	1	0	0	0	0	0
1	2	2100	2	0	1	1	0	1	1	0	1	0	0
1	11	1810	1	0	1	1	0	1	0	0	0	1	0
1	12	1030	1	0	0	1	0	1	1	1	0	0	0
1	1	1709	7	0	0	1	0	0	0	0	0	0	0
1	4	1915	1	0	0	1	0	1	0	1	1	1	0
1	5	600	1	0	0	0	0	1	1	0	0	0	0
1	6	7	2	0	0	1	1	0	0	0	0	0	0
1	11	2135	1	0	0	0	0	1	0	0	0	0	0
1	12	1830	2	0	0	0	0	1	0	0	0	0	0
1	6	1330	3	0	1	0	0	0	0	0	0	0	0
1	10	1500	1	0	1	0	0	0	0	0	0	0	0
1	12	2015	1	0	0	1	0	0	0	0	0	0	0
1	8	203	3	0	1	1	0	0	0	0	0	0	0
1	1	2037	3	0	0	1	0	1	1	1	0	0	0

1	2	1513	8	0	0	1	0	1	0	1	1	0	0
1	1	1340	1	0	1	0	0	1	0	0	0	0	0
1	1	2150	1	0	0	0	0	1	0	0	0	0	0
1	4	1315	2	0	1	0	0	0	0	0	0	1	0
1	9	702	4	0	0	0	0	1	0	0	0	0	0
1	10	2236	3	0	0	1	0	1	0	0	0	0	0
0	7	2205	3	0	0	1	0	0	0	0	0	0	0
1	7	1430	0	0	1	1	0	1	1	0	0	0	0
1	1	1544	0	0	0	1	0	0	0	0	0	0	0
0	3	1922	0	0	1	0	0	1	0	0	0	0	0
1	9	1615	1	0	0	0	0	1	0	1	1	0	0
1	12	1739	1	0	1	0	0	0	0	0	1	0	0
0	8	633	3	0	1	1	0	0	0	0	0	0	0
1	3	1228	2	0	0	1	0	1	0	0	0	0	0
0	3	1207	1	0	0	0	0	1	0	0	0	0	0
1	2	1906	3	0	0	0	0	1	1	0	0	0	0
1	1	2029	4	0	0	1	1	1	0	1	1	1	0
1	4	1755	4	0	1	0	1	0	0	1	0	0	0
1	12	2135	1	0	0	0	0	1	0	0	0	0	0
1	8	554	1	0	0	0	0	0	0	0	1	0	0
1	9	45	2	0	0	0	0	1	0	0	0	0	0
1	11	1630	2	0	1	1	0	0	0	0	0	0	0
1	10	2210	4	0	0	0	0	1	0	0	0	0	0
1	11	1810	3	0	0	1	0	1	0	0	0	1	0
0	1	2008	2	0	0	1	0	1	0	1	0	0	0
1	12	1902	1	0	1	1	0	0	0	0	1	0	0
0	9	2041	2	0	0	0	0	1	0	0	0	0	0
1	3	1935	2	0	0	0	0	1	0	0	0	0	0
1	5	830	2	0	1	1	0	1	0	0	0	0	0
0	9	1850	4	0	1	1	0	0	0	0	0	0	0
1	2	1900	0	0	0	1	0	1	0	0	0	0	0
1	2	1155	1	0	1	0	0	1	0	0	0	0	0
1	6	1218	4	0	1	1	0	0	0	0	0	0	0
1	11	1950	6	0	1	0	0	1	1	0	0	0	0
1	11	1140	1	0	1	1	0	0	1	0	1	1	0
1	11	2040	2	0	1	0	0	1	0	0	0	1	0
0	4	1415	1	0	0	0	0	0	0	1	1	0	0

1	6	1052	0	0	0	1	0	0	0	0	0	0	0
1	11	1830	2	0	0	1	0	0	0	0	0	0	0
1	12	630	2	0	0	1	0	1	0	0	0	0	0
1	11	122	1	0	0	1	0	1	0	0	0	0	0
1	5	1810	1	0	0	1	0	1	1	1	0	0	0
1	5	2250	0	0	1	1	0	0	0	0	0	0	0
1	9	1705	1	0	0	1	0	1	0	0	1	0	0
1	3	922	4	0	1	1	0	0	0	0	0	0	0
1	3	722	2	0	0	1	0	0	0	0	0	0	0
1	3	948	1	0	0	0	0	1	0	1	0	0	0
1	1	1005	3	0	1	1	0	0	0	0	1	0	0
1	1	1517	1	0	0	1	0	1	0	0	0	0	0
0	7	556	1	0	0	1	0	1	1	1	0	1	0
1	6	1655	2	0	0	1	0	1	0	0	1	0	0
1	9	1655	1	0	0	1	0	1	0	0	0	0	0
1	10	1721	1	0	0	1	0	0	0	0	0	0	0
1	3	1930	2	0	0	1	0	1	1	0	0	0	0
1	12	1230	1	0	1	0	0	0	0	0	0	0	0
1	9	1335	4	0	0	1	0	1	0	1	1	0	0
1	9	437	2	0	1	1	0	1	1	1	1	0	0
0	11	2120	2	0	0	1	0	1	0	0	0	0	0
1	11	1140	0	0	1	0	0	0	0	0	0	1	0
1	6	1300	3	0	0	1	0	0	0	0	0	0	0
1	9	1005	5	0	0	0	0	0	0	0	0	1	0
1	3	1252	2	0	0	1	0	0	1	0	1	0	0
1	1	1100	1	0	1	0	0	0	0	0	0	0	0
1	11	1338	2	0	1	0	0	0	0	0	0	0	0
1	4	1130	1	0	1	0	0	1	1	0	1	0	0
0	11	1110	5	0	1	0	0	0	0	0	0	1	0
1	4	1050	3	0	1	0	0	1	0	0	0	0	0
0	5	145	1	0	1	1	0	1	1	1	1	0	0
1	6	615	1	0	1	0	0	0	0	0	0	0	0
1	3	1800	2	0	1	0	0	1	0	0	0	0	0
0	8	2200	2	0	1	0	0	0	0	1	0	0	0
1	4	2040	1	0	0	1	0	1	0	1	0	0	0
1	9	1320	0	0	1	1	0	0	0	0	1	0	0
1	4	1337	1	0	0	0	0	1	0	0	1	1	0

1	4	909	2	0	0	0	0	1	0	0	1	0	0
1	9	1	5	0	0	1	0	1	0	0	0	0	0
1	8	815	0	0	1	0	0	0	1	0	0	1	0
1	2	1845	1	0	1	0	0	1	0	0	1	0	0
1	10	2003	1	0	1	1	0	1	0	0	0	0	0
0	6	2330	0	0	0	1	0	1	0	0	0	0	0
1	10	2031	1	0	0	0	0	1	0	0	0	0	0
0	9	1830	0	0	0	0	0	0	0	0	0	1	0
1	5	838	2	0	1	1	0	1	0	0	0	0	0
0	4	1756	1	0	1	1	0	1	1	0	1	0	0
1	6	2030	0	1	0	1	0	0	0	0	1	0	0
1	7	1030	0	1	1	1	0	0	0	0	0	0	0
1	12	1100	1	1	1	1	0	0	0	0	0	0	0
1	10	1736	0	1	0	1	0	0	0	0	1	0	0
0	8	830	2	1	0	1	0	0	0	0	0	0	0
1	6	827	2	1	0	0	0	1	0	0	0	0	0
0	4	1330	0	1	0	1	0	0	0	0	1	0	0
1	1	1333	2	1	0	1	0	0	0	0	0	0	0
1	8	1922	1	1	0	1	0	0	0	0	0	0	0
1	7	847	0	1	0	1	0	0	0	0	0	0	0
1	11	1510	1	1	0	1	0	0	0	0	0	0	0
1	12	1630	1	1	0	1	0	0	0	0	0	0	0
1	10	1215	1	1	0	1	0	0	1	0	0	0	0
0	10	1100	3	1	1	1	0	0	0	0	1	0	0
1	10	1530	0	1	0	1	0	0	0	0	0	0	0
1	4	0	0	1	0	1	0	0	0	0	0	0	0
1	4	1345	1	1	0	1	0	0	1	0	1	0	0
1	8	1650	4	1	0	1	0	1	0	0	0	1	0
1	11	1300	1	1	1	0	1	0	1	0	1	0	0
1	10	1705	0	1	0	1	0	0	0	0	0	0	0
1	11	1300	0	1	0	1	0	0	0	0	1	0	0
1	11	1300	0	1	0	1	0	0	0	0	0	0	0
1	12	723	0	1	0	0	0	1	0	0	0	0	0
0	12	1600	0	1	0	0	0	1	0	0	0	0	0
1	5	938	0	1	0	0	0	0	0	0	1	0	0
0	6	1154	2	1	1	1	0	0	0	0	0	0	0
1	7	1445	2	1	0	1	0	0	1	0	1	0	0

1	1	1304	1	1	0	1	0	0	0	0	0	0	0
1	8	1900	1	1	1	0	0	1	1	0	0	0	0
1	2	1126	2	1	1	1	0	0	0	1	0	1	0
1	4	1035	4	1	0	0	0	1	0	0	0	0	0
1	8	2110	4	1	0	1	0	0	1	0	1	0	0
1	1	1625	0	1	1	0	0	0	0	0	0	0	0
1	1	1400	0	1	0	1	0	0	0	0	0	0	0
1	2	2140	0	1	0	1	0	0	0	0	0	0	0
1	4	800	0	1	1	1	0	1	0	0	0	0	0
1	5	1245	0	1	0	1	0	0	0	0	0	0	0
0	5	1450	0	1	0	1	1	0	0	0	0	0	0
0	5	1610	1	1	0	1	0	1	0	0	0	0	0
1	7	1900	0	1	0	1	0	0	0	0	0	0	0
1	8	1020	0	1	0	1	0	0	0	0	0	0	0
0	9	1940	0	1	0	0	0	1	0	0	1	1	0
0	9	1700	2	1	0	1	0	0	0	0	1	0	0
0	9	1603	0	1	0	1	1	0	0	0	0	0	0
1	10	1130	1	1	1	0	0	1	0	0	0	0	0
0	11	745	2	1	1	1	0	1	0	0	0	0	0
1	10	1629	0	1	1	1	1	0	0	0	1	0	0
1	10	1430	0	1	0	1	0	0	0	0	0	0	0
1	10	1000	0	1	0	1	0	0	0	0	0	0	0
1	5	1700	1	1	0	0	0	0	0	0	0	1	0
0	1	715	2	1	0	1	0	0	0	0	0	0	0
1	1	1138	0	1	1	0	0	0	0	0	0	0	0
1	3	1130	0	1	0	1	0	0	0	0	0	0	0
1	3	1855	0	1	0	1	0	0	0	0	0	0	0
1	4	1800	1	1	0	1	0	0	0	0	1	0	0
1	4	1345	0	1	1	1	0	0	0	0	0	0	0
1	6	1525	2	1	1	1	0	0	0	0	0	1	0
1	7	1412	1	1	0	0	0	0	1	0	0	0	0
1	7	1050	0	1	0	0	0	1	0	0	0	0	0
0	7	945	0	1	1	1	0	0	0	0	0	0	0
0	7	1215	1	1	1	1	0	0	0	0	0	0	0
0	8	1115	0	1	1	0	0	0	0	0	0	0	0
1	8	1946	1	1	0	1	0	0	0	0	0	0	0
1	9	1545	3	1	0	1	0	0	1	0	0	0	0

1	9	1200	0	1	0	1	0	0	0	0	1	0	0
1	9	1700	0	1	1	1	0	0	0	0	0	0	0
1	10	450	3	1	1	1	0	0	1	0	0	0	0
0	10	1240	2	1	1	1	1	0	0	0	1	0	0
0	12	2300	0	1	1	0	0	0	0	0	0	1	0
0	12	1800	0	1	0	1	1	0	0	0	0	0	0
1	12	1115	0	1	1	1	0	0	0	0	0	0	0
1	12	1445	0	1	0	1	0	0	0	0	0	0	0
1	5	1317	2	1	0	0	0	0	0	0	0	0	0
0	3	845	2	1	1	1	0	0	0	0	0	0	0
1	3	930	0	1	0	1	0	0	0	0	0	0	0
0	4	23	0	1	1	1	0	1	0	0	0	0	0
1	6	1300	0	1	0	1	0	0	0	0	0	0	0
1	7	1930	0	1	1	0	0	0	1	0	0	0	0
1	9	900	1	1	0	1	0	0	0	0	0	1	0
1	9	1759	0	1	1	1	0	0	0	0	1	0	0
1	9	1545	0	1	1	1	0	0	0	0	0	0	0
1	6	126	1	1	1	1	0	1	1	1	1	0	0
0	9	1420	1	1	0	1	0	0	0	0	0	0	0
1	4	1322	0	1	1	0	0	0	0	0	1	0	0
1	1	1155	1	1	0	1	0	0	0	0	0	0	0
0	8	1400	0	1	1	1	0	0	0	0	0	0	0
1	9	1534	1	1	1	1	0	1	0	0	0	0	0
0	12	1100	0	1	0	1	0	0	0	0	0	0	0
0	2	1103	2	1	1	1	0	1	0	0	0	0	0
0	11	500	1	1	1	1	0	0	0	0	0	0	0
1	7	1845	0	1	1	1	0	0	0	0	0	0	0
0	1	2120	2	1	0	1	0	1	0	0	0	0	0
1	7	1610	1	1	1	1	0	0	0	0	1	0	1
0	7	1400	0	1	1	1	0	0	1	0	0	0	0
1	8	0	2	1	0	1	0	0	0	0	0	0	0
1	7	1723	4	1	0	1	0	0	0	0	0	0	0
0	6	1150	0	1	0	1	0	0	0	0	1	0	0
1	10	917	3	1	0	1	0	0	0	0	0	1	0
1	2	1949	1	1	0	1	0	0	0	0	0	0	0
0	3	720	0	1	1	0	0	1	0	0	0	0	0
1	4	1830	0	1	1	1	0	0	1	0	0	0	0

0	7	1212	0	1	0	1	0	0	0	0	0	0	0
0	8	2020	0	1	1	0	0	0	0	0	0	1	0
1	8	1610	0	1	0	1	0	0	0	0	1	0	0
1	6	730	0	1	0	0	0	0	0	0	1	0	0
0	6	600	0	1	0	0	0	1	0	0	0	0	0
1	5	1009	6	1	0	0	0	1	0	0	0	0	0
0	9	2218	2	1	0	1	0	0	0	1	0	0	0
0	3	220	0	1	1	0	0	0	1	0	0	0	0
1	9	1700	0	1	1	1	0	1	0	1	1	0	1
1	6	30	0	1	0	0	0	1	0	0	0	0	0
1	5	905	2	1	0	1	0	1	0	1	0	0	0
1	8	1347	0	1	0	1	0	0	0	0	1	0	0
1	9	243	2	1	0	1	0	0	1	0	1	0	0
1	6	1830	1	1	0	1	0	0	0	0	1	0	0
0	9	1300	0	1	0	0	0	0	0	0	0	1	0
0	11	200	1	1	1	0	0	0	0	0	0	0	0
0	4	1000	0	1	1	0	0	0	0	0	0	0	0
0	8	730	1	1	0	1	0	0	0	0	0	0	0
1	7	615	0	1	1	1	0	0	0	0	0	0	0
0	3	850	4	1	0	1	0	0	0	0	0	1	0
1	5	1000	0	1	0	1	0	0	0	0	1	0	0
0	8	1115	2	1	1	1	0	0	0	0	0	0	1
1	4	1730	0	1	0	0	0	0	0	0	0	1	0
1	6	1500	0	1	0	1	0	0	0	0	0	0	0
1	8	1000	0	1	0	1	0	0	0	0	0	0	0
1	6	1710	0	1	0	0	0	1	0	0	0	1	0
1	8	1930	0	1	0	1	0	0	0	0	0	0	0
1	6	1500	2	1	0	1	0	0	0	0	0	0	0
0	1	1230	0	1	0	1	0	0	0	0	0	0	0
0	8	1858	2	1	0	1	0	0	0	0	0	0	0
1	10	1100	0	1	1	1	0	0	0	0	0	0	0
1	5	1815	0	1	1	1	1	0	0	0	0	0	0
0	8	2315	0	1	1	0	0	0	0	0	0	0	0
1	6	1635	0	1	0	1	0	0	0	0	1	0	0
1	6	1300	0	1	0	0	1	0	0	0	0	1	0
0	9	1715	2	1	0	1	0	1	0	0	0	0	0
0	12	1245	0	1	1	0	0	0	0	0	0	0	0

1	11	1740	0	1	0	1	0	0	0	0	1	0	0
1	8	834	2	1	0	1	0	0	0	0	0	0	0
0	1	1935	0	1	0	1	1	0	0	0	0	0	0
1	5	934	2	1	0	1	0	0	0	0	0	1	0
0	11	1025	2	1	0	1	0	1	0	0	0	0	0
0	9	1106	2	1	1	1	0	0	0	0	0	0	0
0	4	2145	0	1	0	0	1	0	0	0	0	0	0
0	7	1935	0	1	0	1	0	0	0	0	0	0	0
1	7	755	0	1	0	1	0	0	0	0	1	0	0
0	8	1435	0	1	0	1	1	0	0	0	0	0	0
1	10	1628	0	1	0	1	0	0	0	0	0	0	0
1	11	1145	0	1	1	0	0	0	0	0	0	0	0
0	12	1105	2	1	0	1	0	0	0	0	0	0	0
1	7	1000	0	1	0	1	0	0	0	0	0	0	0
1	9	1430	0	1	0	1	0	0	0	0	0	0	0
1	6	1230	1	1	0	0	1	0	0	0	0	0	0
1	4	1600	3	1	0	0	0	1	0	0	0	0	0
1	1	551	2	1	0	0	0	0	0	1	1	0	0
0	10	1820	0	1	0	0	0	1	0	0	0	0	0
0	7	1645	0	1	1	1	0	1	0	0	0	1	0
0	12	130	0	1	0	1	0	1	0	0	0	0	0
0	11	1000	1	1	1	0	0	0	0	0	0	0	0
0	6	1400	0	1	0	1	1	0	0	0	1	0	0
0	5	2145	1	1	0	1	0	0	0	0	0	0	0
0	5	1605	0	1	1	1	0	0	0	0	0	0	0
1	5	1034	2	1	1	1	0	1	0	0	0	0	0
1	5	1300	0	1	0	1	0	0	0	0	0	0	0
0	7	1020	3	1	0	1	0	0	0	0	0	0	0
1	1	1435	0	1	0	0	0	0	0	0	0	1	0
1	9	1632	3	1	0	1	0	0	0	0	0	0	0
1	12	745	0	1	0	1	0	0	0	0	0	0	0
1	4	1312	2	1	0	1	0	0	1	0	0	0	0
1	7	810	0	1	0	1	0	0	0	0	0	0	0
0	6	2323	1	1	0	1	0	0	0	0	1	1	0
0	3	1150	0	1	0	1	0	0	0	0	0	0	0
0	12	1550	0	1	0	1	0	0	0	0	0	0	1
1	8	1115	0	1	0	1	0	0	0	0	0	0	0

0	6	1130	2	1	0	1	0	0	0	0	0	0	0
0	6	1600	1	1	0	1	0	0	0	0	0	0	0
1	9	1131	1	1	1	0	0	0	0	0	0	1	0
1	8	1842	1	1	0	0	0	1	0	0	0	0	0
1	9	1008	0	1	1	1	0	0	0	0	1	0	0
0	6	1018	2	1	1	1	0	0	0	0	0	0	0
1	10	1320	1	1	0	1	0	0	0	0	0	0	0
0	8	945	0	1	0	1	0	0	0	0	1	0	0
1	5	1110	1	1	0	1	0	0	0	0	1	0	0
1	8	2035	2	1	0	1	0	1	0	0	0	0	0
1	8	1100	0	1	0	1	0	0	0	0	0	0	0
1	8	1134	1	1	1	0	0	0	0	0	0	1	0
1	8	2200	4	1	0	0	0	0	0	0	1	1	0
1	8	1320	2	1	0	1	0	0	0	0	0	0	0
1	8	940	1	1	0	1	0	0	0	0	1	1	0
1	8	1830	2	1	0	1	0	1	0	0	1	0	0
1	8	1240	2	1	1	0	0	0	0	0	0	0	0
0	5	1633	0	1	0	1	0	0	0	0	0	0	0
0	10	1510	1	1	0	1	0	0	0	0	0	0	0
0	4	940	0	1	0	1	0	0	0	0	0	0	0
0	6	1415	1	1	0	1	0	0	0	0	0	0	0
1	7	214	0	1	0	1	1	0	0	0	1	0	0
0	8	1030	2	1	0	0	0	0	0	0	1	0	0
1	1	1600	0	1	0	1	0	0	0	0	1	0	0
0	4	2044	1	1	1	1	0	0	0	1	0	1	0
1	8	1800	0	1	1	1	0	0	0	0	0	0	0
1	1	950	0	1	1	1	0	0	0	0	0	0	0
0	10	1040	0	1	0	1	0	0	0	0	0	0	0
1	6	1730	1	1	0	0	0	1	0	0	0	0	0
1	2	1630	0	1	1	1	0	0	0	0	0	1	0
1	9	1556	0	1	0	0	0	1	0	0	0	0	0
0	4	2155	0	1	1	0	0	0	0	0	0	1	0
1	7	1445	2	1	0	0	0	0	0	0	0	0	0
1	12	1045	0	1	1	1	0	0	0	0	0	0	0
1	10	2110	2	1	1	0	0	0	0	0	0	0	0
1	10	1308	1	1	0	1	0	0	0	0	0	0	0
1	8	1635	0	1	1	1	0	0	0	0	0	0	0

0	9	1605	1	1	0	1	0	0	0	0	0	0	0
1	9	1130	0	1	0	1	0	0	0	0	0	0	0
1	6	1830	0	1	0	1	0	0	0	0	0	0	0
1	7	1215	0	1	1	1	0	0	0	0	0	0	0
0	6	1930	1	1	0	1	0	0	0	0	0	0	0
1	2	1512	1	1	0	0	1	1	0	0	0	0	0
0	5	805	1	1	0	1	0	0	0	0	0	0	0
0	5	2013	1	1	0	1	0	0	1	0	1	0	0
0	1	1930	0	1	1	1	0	0	0	0	0	1	0
1	3	1100	0	1	0	1	0	0	0	0	0	0	0
0	3	1605	1	1	0	0	0	1	0	0	0	1	0
1	4	1230	0	1	0	1	0	0	0	0	0	0	0
1	4	1240	1	1	0	1	0	0	0	0	0	0	0
0	5	1720	0	1	1	0	0	0	0	0	0	0	0
1	6	1745	0	1	0	1	0	0	0	0	0	0	0
1	7	1730	0	1	1	1	0	0	0	0	0	0	0
0	8	1415	0	1	1	0	1	0	0	0	0	0	0
1	8	1130	0	1	0	1	0	0	0	0	0	0	0
1	8	1515	0	1	0	1	0	0	0	0	0	0	0
0	9	830	2	1	0	0	0	0	0	0	0	1	0
1	9	1710	2	1	1	0	0	0	0	0	0	0	0
1	9	1100	0	1	0	1	0	0	0	0	0	0	0
1	9	1200	0	1	0	1	0	0	0	0	0	0	0
1	10	1950	4	1	0	1	0	0	0	0	0	0	0
1	10	1815	0	1	0	0	0	0	0	0	0	1	0
1	6	1440	0	1	0	1	0	0	0	0	0	0	0
0	5	1800	1	1	0	0	0	0	0	0	0	1	0
0	1	1500	1	1	0	0	0	1	0	0	1	0	0
1	6	1345	0	1	0	1	0	0	0	1	1	0	0
1	8	1400	1	1	1	1	0	0	0	0	1	0	0
0	9	1845	0	1	0	1	0	0	0	0	0	0	0
0	10	1500	2	1	1	0	0	0	0	0	0	0	0
0	5	1545	0	1	0	0	0	0	0	0	0	1	0
0	4	2330	4	1	0	0	0	0	0	1	1	0	0
0	4	831	1	1	0	0	1	0	0	0	0	0	0
0	7	1600	2	1	1	1	0	0	0	0	0	0	0
0	1	1015	1	1	0	1	0	0	0	0	0	0	0

1	6	924	2	1	0	1	0	0	0	0	0	0	0
1	8	1748	2	1	0	1	1	0	0	0	0	0	0
0	8	1450	0	1	0	0	0	1	0	0	0	0	0
1	5	1000	0	1	0	1	0	0	0	0	0	0	1
1	5	1930	0	1	0	1	0	0	0	0	0	1	0
0	6	1130	0	1	1	1	0	0	0	0	0	0	0
1	9	930	0	1	1	0	1	0	0	0	0	1	0
1	8	1330	0	1	0	1	0	0	0	0	0	0	0
1	11	1	1	1	0	1	0	0	0	0	0	0	0
1	4	1730	0	1	0	1	0	0	0	0	0	0	0
0	8	1530	0	1	1	1	0	0	0	0	0	0	0
1	1	1142	2	1	1	0	0	0	0	0	0	0	0
1	4	1549	1	1	1	1	0	0	0	0	0	1	0
0	2	1423	2	1	0	0	0	1	0	0	0	1	0
0	3	1430	0	1	0	1	0	0	0	0	0	0	0
1	3	1515	1	1	0	1	0	0	0	0	0	0	0
1	3	1630	0	1	1	0	0	0	0	0	0	0	0
1	3	1045	0	1	1	0	0	0	0	0	0	0	0
0	5	1405	0	1	0	1	0	0	0	0	0	0	0
1	5	1505	0	1	0	1	0	0	0	0	0	0	0
0	6	1416	0	1	0	1	0	0	0	0	0	0	0
1	6	1740	0	1	0	1	0	0	0	0	0	0	0
1	8	1215	0	1	1	1	0	0	0	0	0	0	0
1	9	1415	0	1	0	1	0	0	1	0	0	0	0
1	10	2030	3	1	0	1	0	0	0	1	0	0	0
1	11	1941	1	1	0	1	0	0	0	1	0	0	0
1	11	831	3	1	1	1	0	0	0	0	0	0	1
1	10	2025	1	1	0	0	0	1	0	0	0	1	0
0	11	215	0	1	1	0	0	0	0	0	0	0	0
0	1	1110	0	1	1	1	0	0	0	0	0	0	0
0	3	1210	0	1	0	1	0	0	0	0	0	0	0
0	4	1915	0	1	0	1	0	0	0	0	0	0	0
1	2	1310	0	1	0	1	0	0	0	0	0	0	0
0	3	25	2	1	1	0	0	0	0	0	0	0	0
1	5	1045	0	1	0	1	0	0	0	0	1	0	1
0	5	1520	0	1	0	1	0	0	0	0	0	1	0
0	7	1135	2	1	1	1	0	0	0	0	0	0	0

1	7	1506	1	1	0	1	0	0	0	0	0	0	0
1	8	1030	0	1	1	1	0	0	0	0	0	0	0
1	10	1545	0	1	1	0	0	0	0	0	1	0	0
0	5	2253	4	1	0	1	0	0	0	0	0	0	0
1	3	1110	2	1	0	1	0	0	0	0	0	0	0
1	1	1415	0	1	0	1	0	0	0	0	1	0	0
1	5	1510	1	1	1	1	0	1	0	0	0	0	0
1	3	1302	2	1	0	1	0	0	0	0	0	0	0
0	2	934	4	1	1	1	0	0	0	0	0	0	0
1	3	1325	0	1	0	0	0	1	0	0	0	1	0
1	11	1525	0	1	0	1	0	0	0	0	0	0	0
1	9	1430	2	1	0	1	0	0	0	0	0	0	0
0	10	1055	0	1	0	1	0	0	0	0	0	0	0
0	12	1015	0	1	0	1	0	0	0	0	0	1	0
1	7	1616	2	1	0	1	0	1	0	0	0	0	0
1	7	1530	0	1	0	1	0	0	0	0	0	0	0
1	7	1525	2	1	1	1	0	0	0	0	0	0	0
1	11	2115	1	1	1	0	0	0	0	0	0	0	0
0	4	1140	0	1	0	1	0	0	0	0	0	0	0
1	1	1140	0	1	0	1	1	0	0	0	0	0	0
0	6	1530	1	1	1	1	0	0	1	0	0	0	0
0	3	1600	0	1	1	0	0	0	0	0	0	0	0
1	5	1153	0	1	0	1	0	0	0	0	0	0	0
1	7	1900	0	1	0	1	0	0	0	0	0	0	0
1	3	1313	2	1	0	1	0	0	0	0	0	0	0
1	3	914	2	1	0	1	0	0	0	0	0	0	0
1	11	1515	0	1	0	1	0	0	0	0	0	0	0
1	5	1630	1	1	0	0	0	0	0	1	1	0	0
1	6	1616	1	1	1	1	0	1	0	0	1	0	0
1	10	1150	0	1	0	0	0	0	1	0	0	0	0
1	5	738	0	1	0	1	0	0	0	0	0	0	0
0	4	1410	1	1	0	0	0	1	0	0	0	1	0
1	10	1130	0	1	0	1	0	0	0	0	0	0	0
1	3	1354	0	1	0	1	0	0	0	0	0	0	0
0	4	1711	1	1	0	1	0	0	1	0	1	0	0
0	4	1650	3	1	1	1	0	0	0	0	0	1	0
0	2	1100	0	1	0	1	0	0	0	0	0	0	0

1	8	1900	1	1	1	1	0	0	1	0	0	0	0
1	1	1735	0	1	0	0	0	1	0	0	0	0	0
1	2	1330	0	1	0	0	0	0	0	0	0	1	0
0	2	1700	0	1	0	1	0	0	0	0	0	0	0
1	10	1705	1	1	0	1	0	0	0	0	1	0	0
1	1	1116	0	1	0	1	0	0	0	0	0	0	0
1	4	1630	1	1	0	1	0	0	0	0	0	0	0
0	4	1201	2	1	0	1	0	0	0	0	0	0	0
0	7	1723	4	1	1	1	0	1	1	0	1	0	0
1	2	1205	0	1	1	0	0	0	0	0	0	0	0
0	6	1015	1	1	0	1	0	0	0	0	0	0	0
1	1	1100	0	1	0	1	0	0	0	0	1	0	0
0	8	1515	0	1	1	0	0	0	0	0	0	1	0
1	8	1615	0	1	0	1	0	0	0	0	0	0	0
1	12	1330	0	1	0	1	0	0	0	0	0	0	0
1	5	1400	0	1	0	1	0	0	0	0	0	0	0
1	7	1650	1	1	0	1	0	0	1	0	0	0	0
1	8	1835	2	1	0	1	0	0	0	0	0	0	0
0	2	1047	2	1	0	1	0	0	0	0	0	0	0
1	5	1910	0	1	1	0	0	0	0	0	1	0	0
1	12	1424	0	1	1	0	0	0	0	0	0	0	0
0	4	1130	1	1	0	1	0	0	0	0	0	0	0
1	3	1526	0	1	1	1	0	0	0	0	0	0	0
0	8	2000	0	1	0	1	0	0	0	0	0	0	0
1	6	1924	0	1	1	1	0	0	0	0	1	0	0
0	11	1336	0	1	1	0	0	0	0	0	0	0	0
0	3	1820	0	1	1	0	0	0	0	0	0	1	0
0	7	1315	0	1	0	1	0	0	0	0	1	0	0
0	9	1500	1	1	0	1	0	0	0	0	0	0	0
1	11	1350	1	1	1	1	0	0	0	0	0	0	1
1	9	1030	1	1	0	1	0	0	0	0	0	0	0
0	7	1145	1	1	0	1	0	0	0	0	0	0	0
0	6	1800	0	1	0	1	0	0	0	0	0	0	0
1	8	1720	0	1	1	0	0	0	0	0	0	0	0
0	8	1130	0	1	1	1	0	0	0	0	0	0	0
0	12	1600	1	1	0	0	0	0	0	0	0	1	0
0	9	1130	0	1	1	0	0	0	0	0	0	0	0

0	9	1258	1	1	0	1	0	0	0	0	0	0	0
1	9	1215	1	1	0	1	0	1	0	0	0	0	0
1	4	1115	0	1	0	1	0	0	0	0	0	0	0
0	4	1300	1	1	1	1	0	0	0	0	0	0	1
1	10	1234	0	1	0	0	0	0	0	0	0	1	0
1	7	1730	1	1	1	1	0	0	0	0	0	0	0
1	5	1510	0	1	1	0	0	0	0	0	1	0	0
1	10	1052	1	1	0	1	0	0	0	0	0	0	0
1	5	1551	2	1	0	1	0	0	0	0	0	0	0
1	6	1100	0	1	0	1	0	0	0	0	1	0	0
1	9	1017	2	1	0	0	0	0	0	0	0	1	0
0	5	1330	0	1	1	0	0	0	0	0	0	1	0
0	5	1430	0	1	0	1	0	0	0	0	0	0	0
0	7	1330	0	1	1	1	0	0	0	0	0	0	0
1	8	1100	1	1	0	1	0	0	0	0	0	0	0
1	11	1253	1	1	1	0	0	1	0	0	0	0	0
1	8	950	0	1	1	0	0	0	0	0	0	0	0
1	2	1519	1	1	0	1	1	0	0	0	0	0	0
0	8	1434	1	1	0	0	1	0	0	0	0	0	0
0	7	824	2	1	0	1	0	0	0	0	0	0	0
1	9	2152	0	1	0	0	0	0	0	0	0	1	0
1	4	1200	0	1	0	1	0	0	0	0	1	0	0
1	1	1730	1	1	0	1	1	0	0	0	1	0	0
0	4	1805	2	1	0	1	1	0	0	0	0	0	0
0	7	1015	1	1	0	1	0	0	0	0	0	0	0

REFERENCES

Agresti, A., 1990, *Categorical Data Analysis*. Wiley: New York.

Al-Ghamdi, A., 2002, "Using logistic regression to estimate the influence of accident factors on accident severity", *Accident Analysis and Prevention*, 34 729 – 741.

Berry, W., 1993, *Understanding Regression Assumptions*. SAGE Publications Ltd.: Newbury Park.

Carnahan, B., Meyer, G., and Kuntz, L., 2003, "Comparing statistical and machine learning classifiers: Alternatives for predictive modeling in human factors research", *Human Factors*, 45 (3), 408-423.

Carnahan, B., Redfern, M., and Norman, B., 2000, "Designing safe job rotation schedules using optimization and heuristic search", *Ergonomics*, 43 (4) 543-560.

Foster, J., Barkus, E., and Yavorsky, C., 2006, *Understanding and Using Advanced Statistics*. SAGE Publications Ltd.: London.

Genkin, A., Lewis, D., and Madigan, D., 2007, "Large-Scale Bayesian Logistic Regression for Text Categorization", *Technometrics*, 49 (3) 291-304.

Goldberg, D., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley: Boston.

Hanson, J., Redfern, M., and Mazumdar, M., 1999, "Predicting slips and falls considering required and available friction", *Ergonomics*, 42 (12) 1619-1633.

Knecht, W., 2005, Pilot Willingness to Take Off Into Marginal Weather, Part II: Antecedent Overfitting With Forward Stepwise Logistic Regression, Office of Aerospace Medicine Technical Report No. DOT/FAA/AM-05/15, Office of Aerospace Medical Institute, Oklahoma City, OK 73125.

Menard, S., 2002, *Applied Logistic Regression Second Edition*. SAGE Publications Ltd.: Thousand Oaks.

Mesken, J., Lajunen, T., and Summala, H., 2002, "Interpersonal violations, speeding violations and their relation to accident involvement in Finland", *Ergonomics*, 45 (7) 469-483.

Montgomery, D., Peck, E., and Vining, G., 2001, *Introduction to Linear Regression Analysis.* John Wiley & Sons, Inc.: New York.

Pampel, F., 2000, *Logistic Regression A Primer*. SAGE Publications Ltd.: Thousand Oaks.

Pasia, J., Hermosilla, A., and Ombao, H., 2005, "A useful too for statistical estimation: genetic algorithms", *Journal of Statistical Computation and Simulation*, 75 (4) 237-251.

Russell, K., Eccleston, J., Lewis, S., Woods, D., 2009, "Design considerations for small experiments and simple logistic regression", *Journal of Statistical Computation and Simulation*, 79 (1), 81-91.

Tabachnik, B., and Fidell, L., 2001, *Using Multivariate Statistics, 4th Edition.* Allyn & Bacon: Needham Heights, MA.

Yang, L., Dawson, C. W., Brown, M. R., Gell, M., 2006, "Neural network and GA approaches for dwelling fire occurrence prediction", *Knowledge-Based Systems*, 19, 213-219.