EXAMINING THE RELIABILITY OF LOGISTIC REGRESSION ESTIMATION SOFTWARE

by

LIJIA MO

B.Sc. Capital University of Economics and Business, Beijing,
China, 1996.
M.Sc. Humboldt University of Berlin, Berlin, Germany, 2005.

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Agricultural Economics
College of Agriculture

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2010

Abstract

The reliability of nine software packages using the maximum likelihood estimator for the logistic regression model were examined using generated benchmark datasets and models. Software packages tested included: SAS (Procs Logistic, Catmod, Genmod, Surveylogistic, Glimmix, and Qlim), Limdep (Logit, Blogit), Stata (Logit, GLM, Binreg), Matlab, Shazam, R, Minitab, Eviews, and SPSS for all available algorithms, none of which have been previously tested. This study expands on the existing literature in this area by examination of Minitab 15 and SPSS 17. The findings indicate that Matlab, R, Eviews, Minitab, Limdep (BFGS), and SPSS provided consistently reliable results for both parameter and standard error estimates across the benchmark datasets. While some packages performed admirably, shortcomings did exist. SAS maximum log-likelihood estimators do not always converge to the optimal solution and stop prematurely depending on starting values, by issuing a "flat" error message. This drawback can be dealt with by rerunning the maximum log-likelihood estimator, using a closer starting point, to see if the convergence criteria are actually satisfied. Although Stata-Binreg provides reliable parameter estimates, there is no way to obtain standard error estimates in Stata-Binreg as of yet. Limdep performs relatively well, but did not converge due to a weakness of the algorithm. The results show that solely trusting the default settings of statistical software packages may lead to non-optimal, biased or erroneous results, which may impact the quality of empirical results obtained by applied economists. Reliability tests indicate severe weaknesses in SAS Procs Glimmix and Genmod. Some software packages fail reliability tests under certain conditions. The finding indicates the need to use multiple software packages to solve econometric models.

EXAMINING THE RELIABILITY OF LOGISTIC REGRESSION ESTIMATION SOFTWARE

by

LIJIA MO

B.Sc. Capital University of Economics and Business, Beijing, China, 1996.
M.Sc. Humboldt University of Berlin, Berlin, Germany, 2005.

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

Doctor of Philosophy

Department of Agricultural Economics
College of Agriculture

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2010

Approved by:
Co-Major Professor
Allen M. Featherstone

Approved by:
Co-Major Professor
Bryan W. Schurle

Abstract

The reliability of nine software packages using the maximum likelihood estimator for the logistic regression model were examined using generated benchmark datasets and models. Software packages tested included: SAS (Procs Logistic, Catmod, Genmod, Surveylogistic, Glimmix, and Qlim), Limdep (Logit, Blogit), Stata (Logit, GLM, Binreg), Matlab, Shazam, R, Minitab, Eviews, and SPSS for all available algorithms, none of which have been previously tested. This study expands on the existing literature in this area by examination of Minitab 15 and SPSS 17. The findings indicate that Matlab, R, Eviews, Minitab, Limdep (BFGS), and SPSS provided consistently reliable results for both parameter and standard error estimates across the benchmark datasets. While some packages performed admirably, shortcomings did exist. SAS maximum log-likelihood estimators do not always converge to the optimal solution and stop prematurely depending on starting values, by issuing a "flat" error message. This drawback can be dealt with by rerunning the maximum log-likelihood estimator, using a closer starting point, to see if the convergence criteria are actually satisfied. Although Stata-Binreg provides reliable parameter estimates, there is no way to obtain standard error estimates in Stata-Binreg as of yet. Limdep performs relatively well, but did not converge due to a weakness of the algorithm. The results show that solely trusting the default settings of statistical software packages may lead to non-optimal, biased or erroneous results, which may impact the quality of empirical results obtained by applied economists. Reliability tests indicate severe weaknesses in SAS Procs Glimmix and Genmod. Some software packages fail reliability tests under certain conditions. The finding indicates the need to use multiple software packages to solve econometric models.

Table of Contents

Table of Contents v

List of Figures vii

List of Tables viii

Acknowledgements x

1 Introduction 1

2 Literature Review 5

3 Logistic Regression Model 16
- 3.1 Model Assumptions and Definition . . . 16
- 3.2 Model Estimation . . . 17
- 3.3 Model Predictive Ability . . . 19
- 3.4 Model Diagnostics . . . 20

4 Alternative Numerical Algorithms 24
- 4.1 First and Second Derivative Methods . . . 26
- 4.2 Algorithm Comparison . . . 33

5 Software Methods and Reliability 36

6 Software Assessment 41

7 Data Properties and Benchmark Data Generation 44

8 Experiments 48

9 Results and Comments on Software Reliability 51
- 9.1 Dependence on Starting Values . . . 51
- 9.2 Software Reporting . . . 53
- 9.3 Comments on Dependence on Algorithms . . . 56
- 9.4 Comments on Dependence on Observation Number . . . 59
- 9.5 Comments on Dependence on Starting Values . . . 59
- 9.6 Comments on Dependence on Convergence Level . . . 63
- 9.7 Tests Based on Datasets Multi1-7 . . . 64
- 9.8 Summary of Tests Based on Datasets Mvar1-4 . . . 77

9.9 Summary of All Tests Based on Datasets Cuto1-4 89

10 Discussion **92**

11 Conclusion **95**

Bibliography **97**

A List of Acronyms **105**

List of Figures

9.1 The Rank of Software Commands-Minimum Coefficient LRE of Parameter Estimates for Multi7 (on the left with 1000 observations) and Cuto3 (on the right with 5000 observations) (Refer to List of Acronyms) 58
9.2 Comparison of the minimum Coefficient LREs for Different Starting Values for datasets Multi7 and Cuto3 - Refer to List of Acronyms 60
9.3 Comparison of the minimum Standard Error LREs for Different Starting Values for datasets Multi7 and Cuto3 - Refer to List of Acronyms 60

List of Tables

3.1 Prediction Success Table 20

4.1 Comparison of Algorithms 34

6.1 Comparison of Software Commands in SAS 42
6.2 Comparison of Software Commands in Limdep, Matlab, Shazam, R, and Minitab 42
6.3 Comparison of Software Commands in Stata, Eviews, and SPSS 42

7.1 Comparison of NIST Benchmark Datasets 45

8.1 Description of Benchmark Datasets 50
8.2 Description of the Changing the Default Setting 50

9.1 Comparison of Software Commands for Reporting More Significant Digits . . 54
9.2 Comparison of Software Commands in SAS, Limdep, R, Minitab, and Stata Capable of Changing Starting Values 54
9.3 Comparison of Software Commands in SAS, Limdep, Eviews, and Stata Capable of Changing Algorithms 54
9.4 Comparison of Software Commands in SAS, Limdep, Eviews, SPSS, Shazam, and Stata Capable of Changing Convergence Levels 54
9.5 Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in SAS for Testing Multicollinearity 65
9.6 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in SAS for Testing Multicollinearity . . 66
9.7 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in SAS for Testing Multicollinearity . . . 67
9.8 Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in Limdep and Stata for Testing Multicollinearity 68
9.9 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in Limdep and Stata for Testing Multicollinearity 69
9.10 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in LIMDEP for Testing Multicollinearity 70

9.11 Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in Matlab, Shazam, R, Minitab, Eviews, and SPSS for Testing Multicollinearity . . . 71
9.12 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in Shazam, R, Minitab, and SPSS for Testing Multicollinearity . . . 72
9.13 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in R and MINITAB for Testing Multicollinearity . . . 73
9.14 Minimum LREs for Parameter Estimates, Standard Errors, and Log-likelihood Function Changing Default Algorithm in Eviews for Testing Multicollinearity 74
9.15 Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in SAS for Testing Multivariate and Cut-Off Datasets . . . 78
9.16 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in SAS for Testing Multivariate and Cut-Off Datasets . . . 79
9.17 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in SAS for Testing Multivariate and Cut-off Datasets . . . 80
9.18 Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in Limdep and Stata for Testing Multivariate and Cut-Off Datasets . . . 81
9.19 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in Limdep and Stata for Testing Multivariate and Cut-Off Datasets . . . 82
9.20 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in Limdep for Testing Multivariate and Cutoff Datasets . . . 83
9.21 Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in Matlab, Shazam, R, Minitab, Eviews, and SPSS for Testing Multivariate and Cut-Off Datasets . . . 84
9.22 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in Shazam, R, Minitab, and SPSS for Testing Multivariate and Cut-Off Datasets . . . 85
9.23 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in R and MINITAB for Testing Multivariate and Cutoff Datasets . . . 86
9.24 Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in Eviews for Testing Multivariate and Cut-Off Datasets . . . 87

Acknowledgments

I am thankful to the Department of Agricultural Economics and the Department of Economics at Kansas State University and especially, Dr. John Crespi, Dr. Bryan Schurle, and Dr. Hamilton Fout for engaging in my committee of the Ph.D. program. My sincere thanks and deep appreciation are given to my major professor Dr. Allen M. Featherstone and committee member Dr. Jason Bergtold, who have continuously contributed to this dissertation by providing professional advice, valuable suggestions, and constant engagement. I always enjoyed their unhesitating support.

Gratitude is also given to the department head Dr. David Lambert of the Department of Agricultural Economics, Dr. Tian Xia and Dr. Stacy Kovar at Kansas State University and Dr. Kuo-Liang Chang at South Dakota State University for their constructive criticism and moral support at times of ease and hardship.

I am greatly indebted to my family for their spirit and substantial support through the time of my graduate study. They make my life much more plentiful and wonderful.

My special thanks is given to my colleagues of College of Agricultural Dr. Jiarui Li and Dr. Hyun Khang Chang for providing essential support during my hard time, and many graduate students for providing good suggestions.

Chapter 1

Introduction

When doing research, researchers may face discrepancies among the estimation results from different software packages. One of the reasons for these discrepancies may be the numerical accuracy of the software. Testing numerical accuracy can benefit both software users and developers. Benchmark tests used to analyze numerical accuracy of algorithms implemented by software packages are available on the website of the National Institute of Standards and Technology (NIST 2010). Using this information, estimates can be compared to a known level of accuracy for univariate descriptive statistics, analysis of variance, Markov Chain Monte Carlo models, as well as nonlinear and linear regression problems. The performance of a package based on estimating different models allows researchers to understand the strengths and weaknesses of alternative econometric software. Properties of the data may also affect the choice of a suitable package, e.g. the degree of multicollinearity and the number of covariates, because some packages may handle certain statistical issues better than others.

For example, previous research has found that the instability of data is not reflected in small standard errors, even if covariates are highly multicollinear. For nonlinear models, the Jacobian matrix of the $K \times X$ covariate matrix is calculated based on the 1st derivative of the covariate matrix with respect to X. For linear models, conditioning, defined as how close a linear regressor matrix $X'X$ is to singularity and explained in detail in Section 7, is one important factor in accuracy, but it is not the only one. Well-conditioned data are defined with respect to the model and the more pervasive problems of near-multicollinearity are associated with $\widehat{X}'\widehat{X}$ being ill-conditioned, $\widehat{X}$ is the predicted value of X. The inverse of $\widehat{X}'\widehat{X}$ induces numerical instability into estimates of $\widehat{\beta}$. The accuracy of the estimate is a function of the model, data and the algorithm (Altman, Gill and McDonald, 2004).

Benchmark tests are designed to investigate specific weaknesses of statistical software algorithms. The StRD contains multiple problems for testing various aspects of each algorithm. Since the release of the StRD, many reliability tests of statistical packages have been facilitated by running the specified analysis and comparing the results with certified values. It has become a widely used benchmark test for reviewers of software packages (NIST 2010).

A summary of reliability tests (Univariate Statistics; ANOVA; Linear Regression; Nonlinear Regression) have been conducted by researchers for major commercial econometric software packages (Altman et.al., 2004, Chp.3). A reliable package needs to pass all the benchmarks or provide clear error messages when inaccuracy occurs. Out of eleven packages (Limdep, Shazam, TSP, SAS, SPSS, S-Plus, Excel, Gauss, SST, Stata, and Eviews), none of them fails all the benchmark tests (Altman et.al., 2004, Chp.3 Table 3.2). A few packages perform badly on highly difficult nonlinear models (5 out of 11, Eviews, Limdep, Shazam, TSP, SPSS) and in the tails of statistical distributions (4 out of 9) (Gauss 3.2.6, Excel, Shazam, and TSP) (McCullough 1999a p.196, McCullough 1999b p.152-156, p.157, Knuesel 1995).

The type of algorithm chosen and settings used can affect the outcomes of the benchmark tests. When solving the same StRD problem, solvers may yield different results. Nonlinear solvers can be geared to solving nonlinear least squares, maximum likelihood estimation (MLE) and constrained maximum likelihood estimation. The nonlinear benchmark problems in the StRD are formulated as nonlinear least squares problems.

Maximum likelihood problems can be used as generalized formulations in nonlinear regression problems under the assumptions of a known model, a distributional error term and no observation error in the data. Altman et.al. (2004) argue that the nonlinear benchmarks can be used for maximum likelihood search algorithms, but they are not designed specifically for them (pp. 52). The application of the maximization likelihood function to nonlinear least squares was completed by McCullough and Renfro (2000), which explained the numerical sources of inaccuracy of the nonlinear solver. Different software packages deal with nonlinear regression and maximum likelihood estimation differently. Eviews and TSP have separate least squares routines from using the maximum likelihood procedure. Thus separate tests are needed for each maximization routine. Limdep and Shazam combine the two routines (McCullough 1999a pp.198). McCullough and Renfro (2000) suggest that the reason MLE solvers perform worse in solving StRD problems is that nonlinear least squares solvers overtake maximum likelihood solvers by using algorithms with the more restricted structure of the original problem (i.e. Levenberg-Marquardt algorithms), and the MLE reformulation requires additional estimation of the sigma parameter for the variance of the errors.

Logistic regression is one of the most commonly used models using the MLE approach in applied economics. The conventional approach for logistic regression is based on asymptotic maximum likelihood inference. Due to the fact that nonlinear least squares and maximum likelihood estimators are not equivalent in many software packages, the reliability of MLE solvers is not been examined. Greene (2002 Chp.17 pp.496) indicated that the nonlinear least squares estimator ignores the Jacobian of the transformation. In a nonlinear least squares model, nonlinearity in the parameters appears entirely on the right-hand side of the equation in functions of the dependent variable. For the model where parameters are nonlinear functions of dependent variable, the nonlinear least squares estimator is inconsistent due to the ignorance of the Jacobian of the transformation. However, the MLE of this type of model is consistent. If Jacobian terms involve the parameter of the nonlinear dependent variable θ, the least squares is not same as the maximum likelihood.

This study will perform benchmark tests following the reliability evaluation procedures outlined in McCullough (1998, 1999) for logistic regression econometric software packages.

The well-accepted logistic model has important implications for economists in binary choice decisions.

The reliability of logistic regression procedures are tested in SAS, STATA, MATLAB, R, SHAZAM, EVIEWS, MINITAB, SPSS, and LIMDEP, using benchmark datasets developed to test the accuracy and reliability of econometric and statistical software packages in estimating logistic regression models.

Chapter 2

Literature Review

This section analyzes literature related to software reliability. It covers three aspects that impact the reliability of logistic regression models: functional form, cut-off point, and multicollinearity, contributed by authors in the literature in chronological order.

Longley (1967) provides a benchmark for linear regression and compares it to hand calculation results with coefficients produced by the program BMD on IBM 7094; program NIPD on IBM 7094; NIPD' on IBM 7094; ORTHO subroutine of OMNITAB on IBM 7094; IBM 360 CORRE and MULTR subroutines; IBM 1401 Program; IBM 7070/7074 Stepwise Regression Program; and FORTRAN-IV double precision on a IBM 7094 mainframe. The micro-computer linear regression results conclude that these computer programs are not accurate to more than one or two significant digits.

McCullough (1998) provides the foundation for reliability testing that goes beyond the entry-level tests of accuracy for statistical software to provide a collection of intermediate-level tests for linear and nonlinear estimation, random number generation, and statistical distributions. The benchmarks for linear procedures, univariate statistics and analysis of variance are used to test the quality of the software. Before the NIST benchmarks, several researchers provided benchmarks to measure the reliability of software on microcomputers. McCullough (1999b) discusses the reliability of all packages on all data sets listed in Longley (1967). Besides describing the components of the NIST data, McCullough (1999b) raises the importance of testing random number generators, which are important for bootstrapping and Monte Carlo studies, but has not been improved in statistical software.

McCullough (1998) also provides guidelines for assessing the accuracy of statistical distributions, where the crude approximation for calculating p-values requires several digits of accuracy. The computation of a desired probability needs to be accurate up to at least six significant digits, with a relative error smaller than 1E-6. Each distribution has a different numerical underflow limit. The p-values obtained from the lower tail probability may be inaccurate due to cancelation errors. An approach for assessing the statistical distribution is introduced by comparing program results with the Double Precision Cumulative Distribution Function LIBrary (DCDFLIB) and ELV program (for Elementary Distributions, in German: ELementare Verteilungen) critical values.

McCullough (1999b) discusses Marsaglia's (1996) "DIEHARD Battery of Randomness Tests" for random number generators. The DIEHARD tests assume the random number generators tested can produce 32-bit random integers, but a random number generator with enough period length needs to be much greater than $200n^2$ where n is the number of calls to the random number generator. A good random number generator needs to have approximately $n = 1000$ with 2^{31} as its period and will pass almost all 18 randomness tests, some with many variations.

McCullough (1999b) applies his reliability assessment methodology in SAS, SPSS, and S-Plus for estimation, random number generation, and calculation of statistical distributions. Weaknesses are identified in all the random number generators, the S-Plus correlation procedure, one-way ANOVA, and nonlinear least squares problems in SAS and SPSS. SAS correctly solves 20 out of 27, and SPSS solves 26 out of 27 nonlinear problems for the first set of starting values. All random number generators are inadequate because the number of calls made to the random number generator before its repetition is not large enough.

Finite precision cannot be obtained numerically. McCullough and Vinod (1999) argue that some acquaintance with how computers handle numbers is necessary for software users. Computation errors caused by two algebraically equivalent methods may have different effects on computer implementation. Small differences in inputs or algorithms may cause significant differences in estimation results.

McCullough and Vinod (1999) show that the numerical accuracy of econometric software cannot be taken for granted. By surveying five journals that publish reviews of econometric software, they conclude that authors do not consider numerical accuracy. They find 120 econometric software reviews that appeared from 1990-1997 in *the Journal of Applied Econometrics*, *American Statistician*, *Journal of Business Economics Statistics*, *Economics Journal*, and *Econometrica*. Benchmarking has no history in the economics profession. A useful benchmark collection for testing general routines is suggested, especially for economists. More replications are suggested to improve software development, because bugs have been discovered in the full information maximum likelihood estimation of Klein's model, fitting a Cochrane-Orcutt AR(1) correction model, and producing a correlation matrix for independent variables of a multiple regression. The study assesses software packages but leaves the original assessments to the published reviews. The priorities of a software package are providing accuracy, speed and user-friendliness. McCullough and Vinod (1999) argue that improving software is a joint responsibility of the users, developers, and the economics profession.

Nonlinear least squares regression packages obtain different number of accurate digits for estimation for different statistical packages(McCullogh and Renfro, 2000). The FCP (Fiorentini, Calzolari and Panattoni 1996) program, as a benchmark software package, its output is correctly described as possessing both a stopping rule and a test for successful convergence in McCullough and Renfro (1999). Reasons for the differences in results can be explained by the step-length, stopping rule, convergence criterion and method of derivative calculation used by the modelers. Inaccurate estimates are produced by the failure of modelers to implement algorithms properly, e.g. poor implementation of the termination criteria that can be remedied by the gradient evaluation at the maximum. Reliability assessments (McCullough 1999a, b, 2000 and McCullough and Wilson 1999) reveal that false maxima occur when software procedures find a non-maximum solution. Two general results are obtained for nonlinear regressions in SAS, SPSS, and S-plus: analytic derivatives are preferable to numerical derivatives and default estimation is not reliable (McCullough 1999b). Default numerical derivatives must be replaced by analytic derivatives to achieve more digits of accuracy. McCullough (1999a) reports that TSP has the least amount of zero digits of accuracy solutions in 27 StRD tests of nonlinear regression problems with the default options when compared to Limdep, Eviews, and Shazam, indicating TSP has a higher reliability than the rest. Because it takes time to test a package and the general unawareness of pitfalls by the users, sufficient prior testing of algorithms under a variety of different conditions should be widely implemented. The lack of a general appreciation of the circumstances under which an implemented algorithm may fail, contributes to the under-testing of software and the lack of user aids for evaluating these packages.

McCullough and Vinod (2003) propose critical examinations of optimal solutions produced by nonlinear solvers. They study Excel, TSP, Gauss, R and S-Plus and advocate a four-step process for addressing numerical accuracy issues:

1. Examine the gradient by checking if the norm of the gradient is zero. The convergence of the function value and the Hessian should be evaluated at the optimal solution. Relying on nonlinear solver defaults may lead to false convergence.

2. Solution path (i.e. Trace)- Does the solution path exhibit the expected rate of convergence? When the trace of quadratic convergence algorithms exhibit linear convergence in its last few iterations, problems may exist. A solver permitting the user to access the function value along with the estimated parameters can be used to evaluate whether significant changes occur as the solver ceases iterating.

3. Evaluate the Hessian using an eigenvalue-eigenvector analysis to determine if it is negative (or positive) definite for a maximization (or minimization) problem and if it is well-conditioned. If the Hessian is ill-conditioned, the step direction may be unreliable due to round-off error. The largest eigenvalue should be examined as a rule of thumb, because

negative eigenvalues ensure a negative definite Hessian. Bates and Watts (1988, pp. 80) state that the stationary point of a nonlinear minimization problem requires a positive definite Hessian. Multicollinearity analysis is examined using eigen-systems in Vinod and Aman Ullah (1981, sec. 5.3). If the Hessian is ill-conditioned, the difference between the infinitely precise solution and the available precise solution may be large, especially in the direction of the eigenvector associated with the largest eigenvalue of the Hessian at the optimal solution. If the local maximum of the likelihood occurs in a flat region of the log likelihood function surface, then small changes in the log likelihood function may be associated with a large difference in the optimal solution.

Gill et al. (1981, pp. 313) found that if the first three conditions hold, a solution has probably been found, regardless of whether the program has reported convergence. The last step justifies the use of the usual t-statistics for coefficients reported by most packages.

4. The measure of variability of the point estimates are reflected in the standard errors that can be produced by nonlinear routines as t-statistics, or more formally Wald statistics. The validity of profiling the likelihood to assess the adequacy of the quadratic approximation method is based on the shape of the objective function in the neighborhood of the solution. Researchers should examine whether the shape of the objective function is locally quadratic at the solution because a valid Wald test is based on a quadratic functional form.

When a software package declares convergence, researchers should verify the solution to safeguard against false convergence. Software packages differ in both accuracy and the ability to verify a potential solution. A brief review of articles using nonlinear estimation published in the scholarly journals, suggests that published researchers often run solvers with the default settings, that are often found to be less accurate (McCullough and Vinod 2003).

Oster (2002) indicates that the conventional approach for logistic regression is asymptotic maximum likelihood inference. Most software packages use a normal or a chi-square distribution approximation to perform statistical hypothesis testing, under which exact and approximate test results are similar. Conventional approximations do not work well when the data are sparse among categories or unbalanced. The results are not correct when there is a close relation between two or more independent variables (i.e. multicollinearity) or the number of independent variables is too many with respect to the sample size under study. Oster (2002) reviews three statistical software packages (i.e. StatXact, LogXact, and Stata) for analyzing categorical data by using exact methods. StatXact 5, LogXact 4.1, and categorical and nonparametric procedures of Stata 7 are developed for the purpose of performing exact statistical tests for nonparametric statistical inference on either categorical or continuous data. Oster (2003) tests two additional software packages, Testimate 6 and SAS 8.2. Software rating criteria include: installation, interface, ease-of-use, organization and completeness of the documentation, the statistical quality of the documentation, data entry and management, categorical, nonparametric and exact statistical tests, accuracy and reliability aspects. None of the software packages received poor ratings for any of the criteria. Each of them received excellent ratings for at least some of the criteria. Oster (2003) found that researchers performing categorical and/or nonparametric statistical analysis may prefer StatXact due to the limited capability of LogXact. A routine user of logistic regression and/or Poisson regression may choose LogXact because StatXact does not provide relevant capabilities. StatXact and LogXact contain capabilities (exact tests for logistic regression and general linear models) unavailable in other software packages, e.g. Stata and SAS. A number of asymptotic and exact categorical and nonparametric procedures have been added to the most recent releases of SAS and Stata, but Oster (2003) argues that supplementing them with StatXact and LogXact is a wise choice because SAS and Stata are limited in their abilities to do exact tests.

Stokes (2004) investigates potential problems of relying on just one software package in solving a probit model. Quite different results can be obtained using various software packages. The difference can be traced to the default convergence tolerance built into software packages. The nonexistence of MLE probit results can be detected only by changing convergence tolerances. In the iterative process of MLE, if the algorithm can not find an acceptable step to increase the value of the given objective function before the next step, the process may fail to improve the current iteration. A maximum obtained repetitively from increasing the convergence tolerance tests whether the solution is a maximum. It turns out that quasi-complete separation, which occurs when there exists some coefficient vector such that $b'X_i \geq 0$ if $Y_i = 1$; $b'X_i \leq 0$ if $Y_i = 0$ and occurs when a linear function of x can generate almost perfect predictions of Y. Quasi-complete separation causes estimation difficulties only in small sample cases, in which case the MLE may not exist; see Albert and Anderson (1984) for more details. The increased condition number of the Hessian matrix when lowering the tolerance necessitates further checking as to whether the Hessian matrix is becoming singular, a symptom of quasi-complete separation.

Stokes (2004) provides an example of an undetected underlying problem in a nonlinear probit/logit model that had been widely published in various places for many years. The model problems are detected accidently during a routine replication exercise. Only by varying the convergence tolerance is it possible to determine the nonexistence of maximum likelihood solution to the model, as a undetected quasi-complete separation problem. The analysis suggests the constant coefficient assumption implicit in the traditional probit model is not met. A less restrictive model can outperform the probit model with more covariates. Stokes (2004) suggests using more than one software package to solve the same problem and to verify the reliability of the nonlinear solver of probit models in SAS, RATS, Limdep, and Stata.

Keeling and Pavur (2007) compare the relative accuracy of nine software packages (SAS, SPSS, JMP, EXCEL, Minitab, R, Splus, Stata, and StatCrunch) using certified statistical values from NIST. The work contributes by introducing a method for visually plotting the principal components of the correct number of significant digits to reveal how these packages tend to be plotted in a cluster for ANOVA and nonlinear regression. Few substantive improvements have occurred in the area of nonlinear regression procedures in software packages except SAS, whereas the ANOVA procedure has seen the most dramatic improvement in accuracy from the correction of previous inaccuracies. Later versions of SAS, SPSS, Splus, and Excel have demonstrated improvements in their accuracy of statistical analysis compared to prior reliability studies for ANOVA routines.

Keeling and Pavur (2007) indicate that accuracy and convergence are impacted by the default or user selected options in nonlinear regression procedures. The verification of these solutions includes calculating a "profile" function and printing the gradient, Hessian, and function value at every iteration. Finding an appropriate combination of options including convergence criteria, tolerance, method of solution and form of derivative (analytic or numerical) may yield better results than the default settings. A warning or error message is better than a far from optimal solution with no descriptive message for lack of convergence.

Yalta (2007) conducts comprehensive tests of Gauss 8 for estimation, statistical distributions, and random number generation. He shows that statistical distributions and random number generation procedures have serious problems.

Yalta (2007) uses the NIST certified values computed with Bailey's multiple precision FORTRAN preprocessor and NETLIB subroutine library with 500 digits of accuracy for reading and representing data that are rounded to 15 significant digits for numerical tests of linear least squares and 11 significant digits for nonlinear least squares (NIST 2010). In the numerical test of nonlinear least squares, the testing is completed using the "curve fit" package module, which employs the Levenberg-Marquardt algorithm. The primary method for determining convergence is the relative gradient using a default tolerance equal to 1E-5. The regression coefficients and standard errors are calculated in three different ways. For the problems that could not be solved using numerical approximation of derivatives, analytical derivatives were able to solve with more accuracy. The performance of the curve fit module with the default settings is rather poor.

Yalta (2007) performs numerical tests of statistical distributions, using Knsel's (2003) ELV program. The ELV program enables the exact values of nine elementary statistical distributions for probabilities to be calculated as small as 10^{-100}. Generally speaking, GAUSS' performance on statistical distributions is unacceptable, because the comparison between the critical values provided by GAUSS 3.2.6, GAUSS 8, and GNU-R 2.5 and the "exact" counterparts computed by ELV for the F, beta, noncentral chi-square, noncentral F, and noncentral t-distributions shows that the F and Beta distributions are unstable in GAUSS 8, and the noncentrality parameters for the chi-square and the F-distribution need to be revised to comply with the program ELV and GUN-R statistical packages.

Yalta (2007) also mentioned that in comparison with the GNU/Linux version, the Windows version has identical source code for the calculations. For some datasets, the GNU/Linux version of the program fails to converge with analytical derivatives requiring a better set of starting values to converge with the default settings. The producer of Gauss-Aptech's curve fit module was developed for the MS Windows platform and is suspected to be the reason for the greater difficulty in converging for only 11 datasets out of 27 in the GNU/Linux version in comparison with the case 8 out of 27 in the Windows version. Research replication is important in science and needs to cover the data programming code, the software version and the operating system. Algorithms have indicated a trade-off between accuracy and efficiency. Although the high speed is the main advantage of Gauss, modifying the Gauss language to keep its high speed advantage becomes Aptech's main concern. The modification is designed to increase computing costs but gain greater accuracy of the output after iterations by using the premier module "Curve Fit" package rather than using its official "Constrained Optimization" module for solving a constrained nonlinear programming problem.

Bangalore et.al. (2009) assess the tail areas of statistical distributions needed for hypothesis testing, because the accuracy of these areas is important for scientific judgments. In contrast to previous work evaluating statistical softwares' accuracy of tail areas in statistical distributions, Bangalore et.al. (2009) provide guidance on evaluating the accuracy of small tail areas from the cumulative density function (CDF) of the Chi-square and t-distribution, which are important distributions for hypothesis testing. The evaluation is fulfilled by comparing commercially licensed numerical libraries in Java (SSJ, Colt, JSci, Jakarta Math, SOCR, and JMSL), C (GSL) and R (Package Stats) to the accepted standards of ELV and DCDFLIB.

The importance of accuracy of a distribution in the tails and percentiles in statistical distribution using the ELV program by Knuesel (2003) and the DCDFLIB library by Brown et al. (1997) is discussed by McCullough (1998). In Bangalore et.al. (2009), the numerical accuracy compares the tail areas of statistical distributions to exact tail areas of distributions to calculate a Logarithm Relative Error (LRE). The methods used in assessing accuracy are based on guidelines in McCullough (1998) and McCullogh and Vinod (1998). The testing focused on the upper tail of the statistical distribution using the p-values for typical hypothesis testing. When using the complementary lower tail probability to obtain the upper tail, estimating the outcome may be inaccurate because the lower tail probability is near one. At the same time, cancellation errors occurring due to the underflow of floating point representation indicates that complementation (defined as the probability of small tail is equal to one minus the rest area in the CDF) will not work well. Complementation erodes accuracy with an increasing cutoff point. C libraries and R functions have six significant digits of accuracy. Colt is the most accurate of the Java libraries. GSL and R are as accurate as ELV and DCDFLIB and can be used with equal appropriateness as alternative standards of comparison.

Odeh et.al. (2010) investigate the reliability of selected econometric packages (Mathematica, SHAZAM, Microsoft EXCEL, SAS, GAUSS, STATA, MATLAB, R, and GAMS). Accuracy of the estimated results and software operations (including whether errors are correctly identified and reported during estimation) of these packages are studied using the National Institute of Standards and Technology (NIST) benchmark data sets. The accuracies of regression estimates and standard errors are examined for both linear and nonlinear regression problems to investigate econometric software commonly used in economic studies. This article examines whether observed inadequacies in earlier editions of software have been corrected in later versions. They find that the earlier inadequacies have not always been corrected.

Though reliability tests motivate software developers to improve these packages, many deficiencies still exist. Specific studies on the reliability of econometric software for estimating logistic regression models have not been completed to the author's knowledge.

Chapter 3

Logistic Regression Model

The generalized linear model allows for estimating regression models with response data (Y) from the exponential family of distributions:

$$f(y;\theta,\phi) = exp\frac{[y\theta - b(\theta)]}{a(\phi)} + c(y,\phi) \tag{3.1}$$

which includes the normal, binomial, Poisson, geometric, negative binomial, exponential, gamma, and inverse normal distributions (Myers et.al. 2002 pp. 157). With the normal distribution, $\theta = \mu$, $b(\theta) = \frac{\mu^2}{2}$, $\phi = \sigma^2$, $a(\phi) = \phi$, $c(y,\phi) = -\frac{1}{2}[\frac{y^2}{\sigma^2} + ln(2\pi\sigma^2)]$, $E(y) = \mu$, $Var(y) = \sigma^2$, identity link $\eta = \mu$.

For a binary response variable y $P(y = 1) = P$, and y is distributed Bernoulli with $\eta_i = \theta = ln(P/(1 - P))$, $b(\theta) = nln(1 + e^{\theta})$, $\phi = 1$, $a(\phi) = 1$, $c(.) = ln(\frac{n}{y})$.

3.1 Model Assumptions and Definition

Assuming independence, the Generalized Linear Model (GLM) includes three components (McCullagh and Nelder 1983 pp. 19):

1. Random Component: This component is the response variable distribution (or called the error structure). For the logit model, Y follows an independent Bernoulli distribution.

2. Systematic Component: For the logit model, a linear predictor, $\eta_i = \beta' h(X_i) = ln(P/(1 - P))$, is the log odds ratio and P is defined as the expectation of y. $h(X_i)$ is a

function of the explanatory variables X, and $\beta' h(X_i)$ is linear in the parameters.

3. Link Function: A link function connects the linear predictor to the natural mean of the response variable, i.e. $g(\mu_i) = g[E(y_i)] = \eta_i$. In GLMs, it may be any invertible monotonic differentiable function within range (0,1). For the logistic regression model, it is the logit link: $\eta_i = g(\mu_i) = ln\frac{\mu_i}{1-\mu_i}$, making $E(y_i) = \frac{1}{1+e^{-\eta_i}} = F(x)$, $h(\eta_i) = p_i$, and $\eta_i = g(p_i)$. $h(\eta_i)$ is the inverse function of $g(p_i)$ specified as the logit link. The link function maps the unit interval (0,1) onto the real line $(-\infty, \infty)$.

Statistical software programs provide different choices of link functions. For example, Minitab provides binary and ordinary two options for the logit regression. Limdep separates logit and probit model estimation commands.

3.2 Model Estimation

Based on the assumptions and definition, the logit model can be estimated using the maximum log-likelihood method (Gourieroux 2000). The likelihood function for the logit model takes the form:

$$L(y; b) = \prod_{i=1}^{n} \{F(X_i\beta)^{y_i}[1 - F(X_i\beta)]^{1-y_i}\} \tag{3.2}$$

Maximizing the likelihood function is equivalent to maximizing the log-likelihood, where F is the logistic cumulative density function. The log-likelihood function is nonlinear in the parameters.

The log-likelihood can be obtained using the likelihood function in the above equation. That is:

$$\begin{aligned} ln(L) &= \sum_{i=1}^{n} \{y_i ln[F(X_i\beta)] + (1 - y_i) ln[1 - F(X_i\beta)]\} && (3.3) \\ &= \sum_{i:y_i=1} ln[F(X_i\beta)] + \sum_{i:y_i=0} ln[1 - F(X_i\beta)] && (3.4) \end{aligned}$$

When equation 3.3 is strictly concave, a unique global maximum exists. Because the likelihood is a probability between 0 and 1. The $ln(L)$ is always negative (Train 2003 pp.

190).

Differentiating the log-likelihood given by equation 3.3 with respect to the parameter vector β and setting the derivatives equal to zero, yields the gradient or score vector:

$$\begin{aligned} g_i &= \frac{\partial ln(L)}{\partial \beta} = \sum_{i:y_i=1} \frac{f(X_i\beta)}{F(X_i\beta)} X_i^{'} - \sum_{i:y_i=0} \frac{f(X_i\beta)}{1-F(X_i\beta)} X_i^{'} = 0 && (3.5) \\ &= \sum_{i=1}^{n} \frac{(y_i - F(X_i\beta))}{F(X_i\beta)[1-F(X_i\beta)]} f(X_i\beta)X_i^{'} && (3.6) \\ \sum_{i=1}^{n} y_i X_i^{'} &= \sum_{i=1}^{n} F(X_i\beta)X_i^{'} && (3.7) \end{aligned}$$

The final expression for obtaining the parameter set b is based on equation 3.7 or score function, which is a nonlinear equation.

Because the likelihood function is nonlinear in the parameters, closed-form expressions of the MLEs are not available. Thus, the problem must be optimized using iterative numerical methods. The most frequently used methods are discussed in Chapter 4. One such method is the Newton-Raphson (NR) method (McCullough and Renfro 2000 pp. 65, Train 2003 pp. 190). This method finds β^* as follows:

- Begin with the starting value β_0. (The maximum can be traced along the likelihood function surface iteration by iteration until no further increase in the loglikelihood function is found.)

- t iterations later, β reaches its current value β_t.

- After taking a " step", the best value for β_{t+1} will be attained.

- The direction of the step is determined by the gradient g_i (or the slope of the log-likelihood function) evaluated at β_t i.e. g_t, and the length of the iteration is measured by the Hessian H_t, the curvature of the log-likelihood function (i.e. the second derivatives measured at β_t), where $H_t = \frac{\partial^2 log(L)}{\partial\beta\partial\beta'}|_{\beta_t}$.

Fisher information about the unknown parameters β is the expected value of the negative Hessian, which provides information about the curvature and the readiness of discriminating a good parameter estimate, indicated by a small variance.

The MLEs obtained from optimization has the properties:

1. MLEs are either unbiased or asymptotically unbiased with the minimum variance, and additional assumptions are needed for the MLE under certain regularity conditions to ensure this behavior.

2. Global concavity of the likelihood function ensures the existence and uniqueness of MLEs, when the logit model has linear-in-parameters, however, not a guaranteed, functional form (Amemiya 1998 pp.203).

3. Asymptotically normally distributed with mean b and covariance equal to the inverse of the Fisher information matrix (expected value of the negative Hessian) I^{-1}: $\beta_{ML} \sim N(b, \{E[-\frac{\partial^2 log(L)}{\partial b \partial b'}]\}^{-1})$.

4. MLEs form a set of sufficient statistics, i.e. all information about the parameters in the sample is contained in the MLEs.

3.3 Model Predictive Ability

For a binary variable Y with two potential outcomes, the predicted value $\widehat{Y} = F(X_i\widehat{\beta}) = \frac{1}{1+e^{-X_i\widehat{\beta}}}$ may not be consistent with the observed values Y. This inconsistency results in prediction error. Type I prediction error occurs when the wrong observation is predicted incorrectly to be true. Type II prediction error occurs when the correct observation is predicted incorrectly to be wrong. At this point, the cut-off point for classifying the predicted value is very important, because it is used as a threshold to determine the prediction. After the predicted value is compared with the cut-off point, the outcome becomes a binary function of the predicted value. The 2 way contingency table or prediction success table is a way to express this predicted versus observed probabilities, which displays counts of correct and wrong predicted outcomes $\widehat{Y}_{ij}, i, j = 1, 2$ in the cells.

Actual Value	Predicted Value 0 1	Total
0	$\widehat{Y}_{11}$ $\widehat{Y}_{12}$	$\widehat{Y}_{1.}$
1	$\widehat{Y}_{21}$ $\widehat{Y}_{22}$	$\widehat{Y}_{2.}$
Total	$\widehat{Y}_{.1}$ $\widehat{Y}_{.2}$	$\widehat{Y}_{..}$

Table 3.1: *Prediction Success Table*

A typical prediction success table can be expressed in the same way as in Table 3.1.

Some software packages (Limdep, Eviews, and Shazam) provide an actual versus predicted contingency table. Limdep provides an actual probability versus predicted probability as well, but the probability of the cut-off point is 0.5 not the actual success probability, which is changeable to the actual success probability. Eviews uses the actual success probability to determine the percentage of success. Shazam reports this information in Hensher-Johnson Prediction Success Table, which differs from results in other software packages due to its usage of the DFP algorithm. Other software packages output the predicted values after estimation, upon which the prediction success table can be formed based on the predicted value. This prediction success probability is a test statistic depending on the MLE as a function of the probability of conditional mean and its reliability can be evaluated.

3.4 Model Diagnostics

To determine whether a statistical model fits the data adequately, logistic regression uses chi-square statistics to assess the fit of the model. The usual statistics -2 Log Likelihood under null hypothesis provides chi-square statistics. Akaike Information Criterion (AIC) is another way to measure the model fit in many software packages, adjusting the -2 Log Likelihood statistic for the number of terms in the model and the number of observations used. In R, when Log Likelihood is unavailable in the output, AIC becomes a substitute statistic for the Log-Likelihood. If the p-value associated with a chi-square statistic is less than the selected tolerance level, the test rejects the null hypothesis that the model fits the data (The R Book).

McCullough and Vinod (2003) argue that in nonlinear estimation, testing hypotheses, using a Wald-type statistic (with the concomitant asymptotic normal approximation) that is based on a quadratic approximation to the likelihood function maybe invalid. However, one needs to understand the substantial differences between the two approaches. For example, Walter W. Hauck-Allan Donner (1997) mentioned that the Wald test is convenient because it requires fitting the model only under the alternative hypothesis, but the Likelihood ratio test requires fitting the model both under the null and alternative hypothesis. Venables and Ripley (1999, pp. 225) argue that Wald test has a definite advantage over likelihood ratio test, because the maximum likelihood estimation has an iterative nature in approaching an optimum as explained in section 3.2 and Wald test requires a quadratic functional form. Fitting the model twice for the iterative maximum likelihood estimation causes redundant calculations for the computer, which may be unnecessary in the Wald test. McCullough (2004) examined a nonlinear least-squares model with a non-straight line (i.e. not approximately quadratic) profile, and found using Monte Carlo analysis that the likelihood intervals provided better coverage than the Wald intervals. When the Wald statistic interval is valid, so is the likelihood ratio interval, but the Wald statistic interval is valid only if the objective function is quadratic. The likelihood ratio interval is not restricted to the functional form.

Collinearity among the covariates in the logit model is a data problem and not a statistical problem, resulting from ill conditioning in the explanatory variable matrix X. However, collinearity can cause statistical problems. The collinearity problem can be studied by looking at the log-odds equation $ln(P/(1-P)) = \eta_i = \beta' h(X_i)$ (McCullagh and Nelder 1983 pp. 19). The presence of collinearity may degrade the parameter estimates and identifying the affected estimates requires the information only on the data matrix X or the Jacobian. Examining the harm of collinearity requires information on the response variable and/or the variance σ of the error term, because the systematic influence of the explanatory variables on the response variable is swamped by the model's random error term. For a real symmetric matrix $X'X$, its singular values are also its eigenvalues. As the collinearity of columns of any matrix X increases, the matrix tends toward perfect singularity, and eventually, one zero eigenvalue is associated with each pair of exact linear dependency columns. In this way, one small singular value in comparison with the maximum singular value indicates a near dependency among columns of X. The estimated variance of each regression coefficient may be decomposed into a sum of terms each of which is associated with a singular value. This decomposition is a way to determine the degree of near dependencies, indicated by a high condition index. The variance-covariance matrix of the least-squares estimator can be decomposed as $var(b) = \sigma^2 \sum_j v_{kj}^2/\mu_j^2$, where μ_j are eigenvalues, v_{kj} are elements of the orthogonal matrix diagonalizing the squares of the eigenvalue matrix of $X'X$, i.e. the p eigenvectors of $X'X$ associated with nonzero eigenvalues (Belsley, Kuh and Welsch 1980). The eigenvalue decomposition is used to motivate the condition number as a way to detect near-multicollinearity (Spanos and McGuirk 2002).

In any regression, including logistic regression, multicollinearity refers to predictors that are correlated with other predictors. In the logit model, the linear predictor is linear in the parameters and are functions of the explanatory variables. The relationship of the explanatory variables in the above linear model discussion can be applied in evaluating the linear predictor or log odds ratio and therefore the logit model. Moderate multicollinearity may not be problematic. However, severe multicollinearity is problematic, because it can increase the variance of the regression coefficients, making them unstable and difficult to interpret. To measure multicollinearity, examine the correlation structure of the predictor variables and review the variance inflation factor (VIF), which measures how much the variance of an estimated regression coefficient increases if the predictors are correlated (McCullagh and Nelder 1983 pp. 125).

Model validation (Harrell 2001 pp. 91) is used to determine if a model accurately predicts responses on future subjects or subjects not used to develop the model. The quantities used in validation are pseudo R^2 (predictor within-sample), which is a biased measure of model's predictive ability, and the adjusted pseudo R^2, which resolves dramatic overfitting. The unbiased estimates of R^2 can be obtained using data splitting, cross-validation, or the bootstrap. The bootstrap provides the most precise estimates. Two major aspects of predictive accuracy can be assessed by calibration or reliability. Validation of the model ensures that the model is not overfitted. In the logistic regression, calibration can be assessed by plotting Y against the predicted Y. Discrimination is assessed by R^2, which drops in training sample (the full sample less the test sample) from test sample and indicates overfitting. The absolute R^2 in the test sample is an unbiased estimate of predictive discrimination.

Algorithm choice and implementation will impact accuracy. The algorithm is independent of the statistical description of the model and a particular computer program or language. An algorithm includes the choice of mathematical approximations for elements of the model and the method used to find the estimates.

Chapter 4

Alternative Numerical Algorithms

Optimization algorithms are iterative. They begin with an initial estimate of the solution β and generate a sequence of improved estimates (iterations) until they terminate with an "optimal" solution. The strategies used to move from one iteration to the next are different for alternative algorithms. A good algorithm possesses the properties of robustness, efficiency and accuracy, which is defined as the dissimilarity between estimates and output. Robustness requires the algorithm to perform well on a wide variety of problems, for all reasonable starting points. The efficiency criteria indicates the algorithm does not require excessive computer time or storage. There is a trade-off between accuracy and efficiency. In addition, identifying a precise solution without being over-sensitive to errors in the data or arithmetic rounding errors is required for the accuracy criteria. The properties are conflicting: trade-offs exist between convergence rate and storage, as well as robustness and speed (Nocedal and Wright 2000 pp. 8).

An algorithm includes several components: search rule, step length, and termination criteria, and/or stopping rule. Algorithms require information to guide the numerical analysis (Train 2003 pp. 191. and Chapter 8 of Altman et. al. 2004). The stopping rule is needed to determine the necessary conditions for a maximum or minimum. Usually the termination criteria includes one or more of the following conditions:

1. $|Log(L(\beta_{t+1})) - Log(L(\beta_t))| < \varepsilon_s$; that is, successive changes in the log-likelihood function is less than some small value, called the convergence criterion. Finding a flat region of the

Log-likelihood function may indicate an optimum is found, which is associated with a relatively small change in parameter values. When the parameter changes very significantly on a flat surface, the parameter value can not be approached in limited steps. However, when the parameter values change significantly, it becomes a problem indicated by the software output, for example, in Limdep, by outputting "Flat". Software packages can specify this value in the command options, for example, in SAS "Proc Catmod", Epsilon is set at 1E-8 as the default value.

2. $max_i[|\beta^i_{t+1} - \beta^i_t|] < \varepsilon_p$; when the largest successive change in the coefficients is less than the convergence criterion.

3. $g^{'}H^{-1}g < \varepsilon$, where ε is the value of tolerance(), used to compare with the current value of the Hessian-scaled gradient, $g^{'}H^{-1}g$. This is the most commonly used stopping rule.

4. $||g(\beta_t)|| < \varepsilon_g$, sets the magnitude of the gradient to be less than the convergence criterion to check whether the gradient is close to zero. For example, in SAS "Proc Logistic" and "Proc Surveylogistic", "Gconv", the relative gradient convergence criterion, is set at 1E-8 as the default.

The convergence criteria are used to determine if a candidate point is a maximum. Next, the sufficient condition for the maximum is evaluated. The Hessian must be negative definite. This must be done manually and can be only determined locally.

McCullough and Renfro (2000) indicate that some software packages use a single criterion (e.g. stopping rule) for both the stopping rule and convergence criterion. In these cases, the optimization process may find a flat region of the log likelihood function, without checking whether the gradient is close to zero. The negativeness of the Hessian is not guaranteed at the round optimum, possibly indicating a maximum has not been found. McCullough and Renfro (2000) recommend using a software package with an explicit calculation of the Hessian, e.g. Newton Raphson.

Considering that a maximum may occur in a flat region of the parameter space, successive changes in the log-likelihood function could be small at points that are far from the maximum. The gradient may be very close to zero, which is numerically equivalent to zero given the inherent inaccuracy of finite precision computation. The parameters also may not be changing much in such a region.

In the convergence criterion, convergence is defined as the different ratio change of log-likelihood function, as McCullough and Renfro (2000) mentions that it is to locate the flat region of the log-likelihood function between two steps by examining:

$h_t = \beta_t - \beta^*$, where β^* is a local maximum. Using this, the following sequences can be constructed:

Linear: $\frac{||h_{t+1}||}{||h_t||} \leq c$, $h_{t+1} = O(||h_t||)$, e.g. convergence rates e-04, e-04, e-04.
Superlinear: $\frac{||h_{t+1}||}{||h_t||} \rightarrow 0$, $h_{t+1} = o(||h_t||)$, e.g. convergence rates e-03, e-04, e-06.
Quadratic: $\frac{||h_{t+1}||}{||h_t||^2} \leq c$, $h_{t+1} = O(||h_t||^2)$, e.g. convergence rates e-02, e-04, e-08.

If $h_t = log(L(\beta_t)) - log(L(\beta^*))$, where $log(L(\beta^*))$ is the value at the maximum, the rate of convergence applies to the value of the objective function.

First derivative methods and second derivative methods can both be applied in finding an optimum solution. The main difference between them is that first derivative methods do not use Hessian matrix and second derivative methods need to compute the Hessian matrix, or an approximation to it. Therefore, first derivative methods are usually, but not always faster than second derivative methods in finding the optimal solution. In second derivative methods, the Hessian is expensive to compute, but results may be more reliable.

4.1 First and Second Derivative Methods

1. Newton-Raphson (NR): According to numerical theory, parameter estimation can use the Newton-Raphson method with the second-derivative matrix being replaced by its expected value, where the search or iterative rule is derived from the $log(L)$'s second-order Taylor's series expansion around β_t: $Log(L(\beta_{t+1})) = Log(L(\beta_t)) + (\beta_{t+1} - \beta_t)g_t + \frac{1}{2}(\beta_{t+1} - \beta_t)^2 H_t$. This gives the first-order Taylor approximating score function $g(b)$ about the starting value

β^0,

$$g(\beta) = g(\beta^0) - I(\beta^0)(\beta - \beta^0) \tag{4.1}$$

$$\beta = \beta^0 + I^{-1}(\beta^0)g(\beta^0) \tag{4.2}$$

where $I(\beta^0)$ denotes the observed information matrix evaluated at β^0.

In an iterative manner, the i+1th estimate is obtained based on: $\beta^{i+1} = \beta^i + I^{-1}(\beta^i)g(\beta^i)$. The algorithm iterates until the -2 Log Likelihood changes by only a small amount from the previous iteration, when any of the stopping rules mentioned above are satisfied (Harrell 2001 pp. 192).

The value of β_{t+1} maximizing $Log(L(\beta_{t+1}))$, i.e. a search rule can be found by setting first derivative to zero:

$$\frac{\partial Log(L(\beta_{t+1}))}{\partial \beta_{t+1}} = g_t + H_t(\beta_{t+1} - \beta_t) = 0 \tag{4.3}$$

$$\beta_{t+1} = \beta_t + (-H_t^{-1}g_t) \tag{4.4}$$

$-H_t^{-1}$ valued at β^0 is equal to $I^{-1}(\beta^0)$ in equation 4.2. When the log-likelihood function is quadratic, it takes only one step to reach the maximum. However, because most log-likelihood functions are not quadratic, it takes more than one step to reach the maximum. A successful algorithm uses a local quadratic approximation to capture the shape of the function to estimate the next step. To allow for the possibility of stepping past the maximum to a lower $Log(L(\beta_{t+1}))$ for a global maximum, the second term on the right hand side is multiplied by the step size λ, which becomes: $\beta_{t+1} = \beta_t + \lambda(-H_t^{-1})g_t$. If the approach ends up with a tiny λ, it safe-guards that a different iteration procedure-"line search" should be used. If a rising λ is arrived at in approaching the maximum, it reduces the number of iterations needed. Adjusting the value of λ instead of recalculating the H_t and g_t, can quicken the search for the maximum in each iteration. Line search starts by fixing a search direction d_t and then identify an appropriate distance λ to move along (Train 2003 pp. 194; McCullough 2004 pp. 201). The speed of convergence is quadratic.

If the log-likelihood function is globally concave, the Newton-Raphson method guarantees an increasing likelihood function at the next iteration with $-H_t^{-1}$ being positive definite. To guarantee an increase at each iteration even if in convex regions of the log-likelihood function are encountered, a reconstruction of $-H_t^{-1}$ is necessary to make calculation easier and ensure that the Hessian is positive definite (Train 2003 pp. 196).

The estimated covariance matrix of the parameter estimators using the Newton-Raphson method is based on the observed information matrix.

2. Weighted Least Squares Estimate

An alternative estimator to MLE as a solution to the score equation is weighted least squares. The Newton-Raphson method can be used to solve the score equation. The first order Taylor expansion around the parameter vector β can be rewritten to arrive at: $y_i - \mu_i \approx var(y_i)(X_i\beta^* - X_i\beta)$, with $var(\eta_i^*) \approx \frac{1}{var(y_i)}$, $\eta_i = X_i'\beta$, where β^* is the value that solves the score function, η_i^* is the value of η_i evaluated at β^*. $\mu_i = n_i p_i$ is the binomial mean and y_i is the binomial response variable. Assuming m denotes the number of independent binomial observations, $V = diag(e^{X_i\beta})$, giving:

$$\begin{aligned}
\sum_{i=1}^{m} var(y_i).(\eta_i^* - \eta_i) &= 0 && (4.5)\\
X'V(\eta_i^* - \eta_i) &= 0 && (4.6)\\
X'V(\eta_i^* - X\beta) &= 0 && (4.7)\\
\widehat{\beta} &= (X'VX)^{-1}X'V\eta^* && (4.8)
\end{aligned}$$

Standard errors of the estimated coefficients can be estimated using $Cov(\beta) = (X'\widehat{V}X)^{-1}$, which replaces β with MLE b in V.

$y_i/n_i - P_i \approx (\partial P_i/\partial\eta_i)(\eta_i^* - \eta_i)$; $\eta_i^* \approx \eta_i + (y_i/n_i - P_i)(\partial\eta_i/\partial\pi_i)$; $b = (X'VX)^{-1}X'Vz$.

IRLS and MLE are equivalent asymptotically. This can be proven from the following steps:

1). The IRLS algorithm (or maximum quasi-likelihood) based on the Newton-Raphson method uses ordinary least squares to obtain an initial estimate of β_0 2). uses β_0 to then estimate V and π; 3). lets $\eta_0 = X\beta_0$; 4). bases z_1 on η_0; 5). obtains a new estimate β_1, and iterates until convergence (Myers, Montgomery and Vining 2002 pp. 326).

3. Quadratic Hill-Climbing (Goldfeld-Quandt): As a straightforward variation of Newton-Raphson, this method modifies the Newton-Raphson algorithm by adding a correction matrix to the Hessian. The updating algorithm is given by

$$\beta_{t+1} = \beta_t - \widetilde{H_t}^{-1} g_t \tag{4.9}$$

$$-\widetilde{H_t} = -H_t + \alpha I \tag{4.10}$$

where I is the identity matrix and α is a positive number chosen by the algorithm when refitting. The asymptotic standard errors are always computed from the unmodified (Gauss-Newton) Hessian after convergence. This modification will push the parameter estimates in the direction of the gradient vector. The idea is that when the solution is far from the maximum, the local quadratic approximation to the function may be a poor guide to its overall shape, so one may be better off following the gradient. The correction may provide better approximation at locations far from the optimum, and allows for computation of the gradient vector in cases where Hessian is near singular with a improved convergence rate (Eviews 3 Manual User's Guide 2002 pp. 620).

4. Method of Scoring or Fisher Scoring (or Iteratively Reweighted Least Squares (IRLS): This method is performed by taking the expectation of the Hessian conditional on x, to form the expected information matrix (EIM). This method just replace the Hessian matrix with I_t^{-1} using the score function in the estimator. It is an optimization of the deviance not of the log-likelihood. The deviance is defined in Goodness-of-fit tests, one of which uses the likelihood ratio criterion to define this concept. Deviance allows one to determine if the fitted logistic model is significantly worse than the saturated (or full) model. $E(y_i) = P_i, i = 1, 2$ the saturated model has y_i the binary observation as the estimator $\widehat{P}_i$. The saturated model requires estimation of two parameters independent of regressor information. The deviance is defined as twice the difference between the maximum attainable log likelihood (i.e. the full model $L(P)$) and the log likelihood of the model under consideration (i.e.

the reduced model $L(\beta)$). The deviance is often used as a measure of goodness of fit. Thus, the deviance is $D(\beta) = -2ln[L(\beta)/L(P)]$, where $L(\beta)$ is the likelihood for the fitted logistic model with parameter replaced by MLE and $L(P)$ is the likelihood for the saturated model with 2 parameters estimated as before. For formal testing, one makes use of the fact that asymptotically $D(\beta) \sim \chi^2_{2-p}$, where p is the degree of freedom. The maximum attainable log likelihood is achieved with a model that has a parameter for every observation. The estimated covariance matrix of the parameter estimators in Fisher scoring is based on the expected information matrix. It has good asymptotic properties. The weighted least-squares estimation is identical to the method of scoring (Gourieroux 2000 pp. 18). In the case of a binary logit model, the observed and expected information matrices are identical and resulting in identical estimated covariance matrices for both Newton-Raphson and Fisher Scoring algorithms, although empirically, this does not always work out. The equivalence of IRLS and MLE has been verified to obtain a significant estimated of β for infinite observations.

5. Berndt, Hall, Hall, Hausman (BHHH): Called the Gauss-Newton for general nonlinear least squares with a line search, this method is based on the Newton-Raphson algorithm. The BHHH method can take more iterations to obtain the optimum than the Newton-Raphson method, but each iteration is simpler computationally and less time-consuming (Fiorentini et.al. 1996). In comparison with the Newton-Raphson, the $-H_t^{-1}$ is replaced by B_t. The score is defined as $S_n(\beta_t) = \frac{\partial lnP_n(\beta)}{\partial\beta}|_{\beta_t}$. P_n is the independent probability of each observation. Thus, the gradient is the average score $g_t = \frac{\sum_n S_n(\beta_t)}{N}$. At the optimal parameter values, the average score is zero. The outer product of the Jacobian of the score is a $K \times K$ matrix, where K is the dimension of parameter β. $B_t = \frac{\sum_n S_n(\beta_t)S_n(\beta_t)'}{N}$ approximating I_t^{-1}, is the covariance matrix of scores averaging the outer product of the gradient (OPG), which is necessarily positive definite. When the negative of the Hessian is replaced by summing up OPG vectors for each observation's contribution to the objective function, the approximation is asymptotically equivalent to the actual Hessian evaluated at

the optimal parameter values.

At the maximizing values of the parameters, B_t is the variance of scores. Similar to the Hessian, the variance of scores provides a measure of the curvature of the log-likelihood function. The relationship between the variance of the scores and the curvature of the log-likelihood function is stated as the information identity: the covariance of the scores evaluated at the true parameters is asymptotically equivalent to the negative of the expected inversed Hessian. This identity holds at the maximum of the log-likelihood function. When the values of β are farther from the true values, the approximation is poorer.

6. Berndt, Hall, Hall, Hausman 2 (BHHH-2) At non-optimal parameter values, the log likelihood is not at a maximum and the average score is non-zero. This variant of the BHHH is determined by subtracting the mean score before taking the outer product. The matrix $W_t = \sum_n \frac{(S_n(\beta_t)-g_t)(S_n(\beta_t)-g_t)'}{N}$ represents the covariance of the scores around their mean. When W_t is taken as B_t, the iteration reaches the maximum for the parameters. W_t and B_t are identical at the optimal value of β. The maximization procedure becomes: $\beta_{t+1} = \beta_t + \lambda(W_t^{-1})g_t$. λ is a line search and is used to catch the difference between W_t and B_t without actually adjusting the value of W_t.

7. Steepest Ascent. When the identity matrix is used in place of W_t^{-1}, an increasing iteration is guaranteed: $\beta_{t+1} = \beta_t + \lambda g_t$. The gradient g_t provides the direction of greatest increase in the log-likelihood function. It moves from an initial stating point β_0 in the direction of g_t. Usually, the step size is small. This method requires only the function and the gradient values to conduct a search. The speed of the convergency is slow, which needs more iterations than other algorithms, exhibiting linear convergence. As the algorithm approaches the optimum, the procedure will produce shorter moves with a zig-zag behavior (Nocedal and Wright 2000 pp. 43).

8. Broyden-Fletcher-Goldfarb-Shanno (BFGS) is a Quasi-Newton Method or Variable Metric, Secant Method, which uses a sequence of approximations to the Hessian matrix. At the $t + 1$ iteration, the approximation of the Hessian matrix depends on the approximated

Hessian matrix at tth iteration and on the gradient at the current iteration g_{t+1}. The quantity $(g_{t+1} - g_t)^T(X_{t+1} - X_t)$ approximates the curvature and the H_{t+1} determined in this way is not unique. The approximation of the Hessian is symmetric and positive definite. It is generally agreed that the best overall performance is achieved by the BFGS update: $H_{t+1} = H_t - H_t s_t (H_t s_t)^T / s_t^T H_t s_t + y_t y_t^T / y_t^T s_t$, where ($s_t = X_{t+1} - X_t$, $y_t = g_{t+1} - g_t$). BFGS update is a rank-two modification of the current approximated Hessian and is superior in computation without a precise reason (Fraley 1989). BFGS is the default algorithm for the optimization in many software packages. The underlying difficulty is the use of only gradient information, which may not detect a saddle-point (McCullough and Renfro 2000 pp. 71).

The update preserves positive definiteness of H_t whenever $y_t^T s_t$ is positive, a condition that automatically holds in a line search method satisfying $|g(x_t + \alpha_t p_t)^T p_t| \leq -\eta g_t^T p_t$ as the termination criterion. (α_t is the sufficiently small value as a step-length, p_t is the direction of an iterative process, $\eta \in$[0,1) keeps the step-length bounded away from zero by forcing it to approximate a local minimum of F along the direction of p_t, and g_t is the gradient.)

9. Davidon-Fletcher-Powell (DFP) is a Conjugate Gradient method that uses information at more than one point β_t on the likelihood function to calculate the approximate Hessian. Information at one point β_t can determine the shape of the function. In the place of equation 9, a substitution of H_t by Q_t, and Q_t is constructed so as to be negative definite at any β_t ensuring the convergence properties to be independent on the starting values. The main point of the substitution is to update the matrix Q_t by a low-rank matrix iteratively and eventually attain the convergence to the Hessian. To solve the problem of negative definiteness center around replacing Hessian in Newton-Raphson with a matrix Q_t to form the quasi-Newton (variable metric) method, where Q_t is constructed by the function F and g_t. H_t is approximated in several iterations by Q_t at the convergence. At each iteration, Q_t is updated by $Q_{t+1} = Q_t + U_t$. U_t is a low-rank updating matrix. Both approaches are based on rank two (U_t has rank two) updates, and build up curvature information slowly.

When convergence is not attainted before the iteration ends, the coefficient estimates may be good, but the standard errors will not be as good, because Q_t is not a good approximation to H_t. The arc Hessian can be used to approximate the Hessian, which reflects the change in gradient that occurs for accrual movement along the log-likelihood function curve. The Hessian reflects the change in slope for infinitesimally small steps around the point β_t. Different rules for constructing U_t make the BFGS different from DFP (see Fletcher 1987 pp.55 for more details). The later is a refinement of DFP which adds another rank one term (thus making in a rank three update)(McCullough Renfro 2000). Broyden-Fletcher-Goldfarb-Shanno (BFGS) for the detail on how to calculate Q_t. The speed of convergence is superlinear.

10. Grid Search simply computes the objective function on a grid of parameter values and chooses the parameters with the highest values. It is computationally costly especially for multi-parameter models. It is used together with the other derivative methods to search for the optimum solution in some statistical packages(Eviews 3 Manual pp. 620).

4.2 Algorithm Comparison

See Table 4.1 for detail on comparison among Newton Raphson, Berndt-Hall-Hall-Hausman, Berndt-Hall-Hall-Hausman2, Steepest Ascent, Davidon-Fletcher-Powell, and Broyden-Fletcher-Goldfarb-Shanno. The Gauss Newton's method is a special case of the Newton-Raphson method. The first order condition of log-likelihood function with respect to the parameter vector b can be approximated by a first-order Taylor expansion around an initial value b_0. The b_{n+1} can be solved iteratively based on b_n (Gourieroux 2000 pp. 17).

Generalized or weighted least squares are very important methods for estimating the parameters in GLM. It is shown that the (iteratively) reweighted least squares and maximum likelihood score equation are asymptotically the same (Myers 2002 pp. 106 and pp. 325). In weighted least squares, $H = -X^{'}W_0X$(minitab), where X is the design matrix. W_0 is a diagonal matrix with its ith diagonal element equal to weight on log-likelihood. (SAS 9.

Table 4.1: *Comparison of Algorithms*

Newton Raphson	BHHH	BHHH2	Steepest Ascent	DFP and BFGS
PROS				
More dependent on choice of starting value. than Quasi-Newton	B_t is calculated faster than H	$W_t$is necessarily positive definite, which ensures an increasing log-likelihood across iterations.	Works best when the starting point is far from the optimum.	Uses information at several points to obtain curvature of the function.
Calculates Hessian directly, standard errors are more exact.	B_t necessarily positive definite	Information identity holds, when the model is specified at the true parameters.	Numerically evaluates only function and gradient. No use of Hessian.	Works well even if the function is nonquadratic. DFP:Conjugate Gradient BFGS: Quasi-Newton
Convergence				
Speed: quadratic in final iterations	quadratic	quadratic	linear	superlinear
CONS				
Calculation of H is computationally-expensive	When the iteration is far from the maximum, it gives small steps that increases the log-likelihood in small increments.	W_t is not the variance of scores.	Converges slower than BHHH and BHHH2.	Uses approximate Hessian. BFGS:Best for median-sized problems. DFP: best for larger problems.
Does not guarantee an increased iteration if the log -likelihood is not globally concave. If the function is highly non-quadratic, it does not work well.	Uses outer product of gradient to approximate negative of Hessian.	Approximates the inverse Hessian by demeaning OPG with the gradients.At the places other than maximum, W_t is not same as inverse Hessian.	Requires more iterations. Zigzaging close to optimum	Standard errors are not as good as actual using H^{-1}, if convergence is not attainted.

GENMOD command reference).

Algorithm robustness depends not only on the problem, stopping rules, and step length, but on the amount of derivative information required on the gradient, or gradient and Hessian, and the method of determining derivatives (i.e. numerical or analytic) (McCullough and Renfro 2000).

Chapter 5

Software Methods and Reliability

Learning computer software languages and how the computer manipulates real numbers, provides assistance for evaluating software reliability. Computational results rely on both hardware and software, distinguishing their difference and impacts on reliability can provide value in assessing software reliability.

A computer uses finite discrete approximation to deal with real numbers instead of using real numbers themselves; therefore, real number arithmetic is the heart of software computing and understanding the function of algorithms. Reliability involves the interplay between the hardware, compiler, and algorithms. Sawitzki (1994a) found that truncation error is the only error caused by limitation of the software. The study shows that a relative change in the data on order of 10^{-3} in his example can produce a change of order of 10^{-4} in the coefficients of the normal equation. Truncation error may be considered as an approximation error. When iterative algorithms are subject to truncation error, the algorithm will provide the correct answer after an infinite number of iterations. However, the iterations of most MLE algorithms converge after a finite time of iterations, which makes investigating incorrect answers caused by the truncation error in finite MLE iterations important.

Monahan (2001) indicates that there are two ways to represent numbers in computers: one is fixed numbers for representing integers, and another is using floating point numbers for representing real numbers. Early computers used fixed point representation and some used floating point. By the late 1950s, the range limitations of fixed point restricted its practice, and floating point was shown to be more versatile and efficient. The fixed point representation had been criticized for using bits to store the exponent, which is seen as a waste when they could be used to extend the precision of the significand.

In fixed point representation, three fields are used to store a number: one 1-bit field for the sign of the number, a field of bits for the binary representation of the number before the binary point, and one field of bits for the binary representation after the binary point. The fixed point system is limited by the size of the numbers it can store, therefore it is rarely used for numerical computation (Monahan 2001).

Floating point representation is based on exponential (or scientific) notation, where a nonzero real number x is expressed in decimal as $\pm S \times 10^E$, where $1 \leq S < 10$, and E is an integer. S and E are called the significand and the exponent. A normalized number can be stored in three fields: one for the sign, one for the exponent E and one for the significand S. A real number is called a floating point number if it can be stored exactly on the computer using the floating point representation. Base 2 is preferred to base 10 (Monahan 2001).

Overton (2001) indicates the precision of the floating point system is the number of bits in the significand (including the hidden bit). Usually precision is denoted by p. In a computer system with $p = 24$ (i.e. single precision), 23 bits are used to store the fractional part of the significand with 1 leading hidden bit. Storage ϵ (machine epsilon) is the unit of roundoff, which is the gap between 1 and the next largest floating point number.

Any positive number z can be expressed as a base-B number with a set of digits $\{0, 1, B-1\}$, for example, with base B=2, the set of digits being used to express any positive number is just $\{0, 1\}$. Using a converging series, a real number can be expressed as:

$$z = a_k B^k + \ldots a_1 B^1 + a_0 + a_{-1} B^{-1} + a_{-2} B^{-2} + \ldots \tag{5.1}$$

The coefficients a_j are integers in the set of digits $\{0, 1, B-1\}$, which give the expression of the real number as the list of digits: $z = (a_k \ldots a_1 a_0 a_{-1} a_{-2} \ldots)_B$. The radix point or base point denotes the period in this representation. The following example shows how the base

2 and base 10 representations are inter-changeable. When z is expressed in base 2, $a_3 = 1$, $a_2 = 0$, $a_1 = 1$, and $a_0 = 1$ in equation 5.1, the representation is 1011. If using base 10 as the base number, $a_1 = 1$, and $a_0 = 1$ in equation 5.1, the representation is 11. The intermediate expressions are derived based on equation 5.1.

$z = 1011_2 = 12^3 + 02^2 + 12^1 + 12^0 = 11_{10} = 110^1 + 110^0$

Because the exponent E is bounded for floating point numbers, a finite number of values with floating point notation compose a finite set of representable numbers V, which can be used to approximate any real number. One can represent a real number z lying between two representative numbers, (S, E, F) and $(S, E, F + B^{-d})$. For example, a real number 3.235710^{23} becomes a choice between (+, 24, .323) and (+, 24, .324) depends on the methods of rounding being chosen. This type of error due to finite precision is rounding error, and it is a function of hardware. Rounding to the closest digit gives (+,24, .324); and chopping or dropping the last coefficient gives (+, 24, .323). The choice between rounding and chopping (or dropping) affects the result of analysis on accuracy.

Second, there are multiple representations for the same number in floating point notation. The value 6 can be expressed by (+,1,.6000) or (+,2,.0600), or (+,4,.0006). The first expression is preferred due to its correct conveyance of the accuracy of the number. The last can express any number in the interval (5.5, 6.5), for example, 5.8 can be expressed as (+,4,.0006) by rounding up the last number after the decimal point to 1. The representations with the most significant (the leading or leftmost) digit is nonzero are called normalized. The inaccuracy of the computation is caused by the misuse of the interchangeable un-normalized floating point notations in computer memory.

Most microcomputers and workstations follow the IEEE binary floating point standard, which was a milestone in computer arithmetic in recent years. A single-precision floating point number is typically stored using 32 bits. A computer uses 23+1 bits to store the significand S, 8 bits for the exponent, unit roundoff is $2^{-24} \approx 5.96 \times 10^{-8}$, with range of $10^{\pm 38}$. A double-precision floating number uses double the storage of a single-precision number, i.e. 64 bits. The computer uses 52+1 bits to store the significand, 11 bits for the exponent, $2^{-53} \approx 1.11 \times 10^{-16}$ for unit roundoff with range of $10^{\pm 308}$ (Higham 2002). "Double precision" has been adopted by most software package and provides greater accuracy by avoiding the perils of floating point arithmetic and roundoff error.

Quadruple precision uses 128 bits as storage for floating point numbers. Higher precision arithmetic is implemented using arrays of double precision variables. A "Double-double" format uses two double precision variables to represent a variable with twice the precision, leading digits in the first array, and trailing digits in the second array (Overton 2001 pp. 23).

The arrangements for storing single and double precision numbers are different. Thus, it takes twice the storage to store a double precision number. In addition, it takes three or four times longer to finish a multiplication or division with double precision numbers.

McCullough (1998) points out that single precision is only accurate up to six or seven digits, while double precision has 15 to 16 digits of accuracy. McCullough (1998) also emphasizes the importance of the combination of using appropriately accurate programs with a good algorithm in passing reliability tests. The infiniteness of the floating point expressions of some numbers leads to either rounding or truncation errors at the last stored bit (Altman, Gill, and McDonald 2004 pp. 25).

In addition to rounding and overflow, operations in floating point are problematic due to underflow. Underflow occurs when a number is smaller than the smallest value capable of being represented by the computer: the unit of roundoff or machine epsilon ϵ. In the double precision computer system, when the number is smaller than 2^{-52}, underflow will occur. The smallest quantities between machine epsilon and one are subnormal numbers (McCullough and Vinod 1999 pp. 643 and Altman et. al. 2004 pp. 24), and when they are added, they do not have full precision. If two nearly equal numbers are subtracted, cancelation may occur, and as a result, rounding error is left (Higham 2002, pp. 493).

As a result of the inaccuracy, floating point arithmetic does not obey the algebraic associative law: if a list of numbers is summed up in different order, a different sum may be computed. In floating point expression, the addition of 1 to 10000 (+,5,.1000) does not increase the sum, and the sum is (+,5,.1000). Similarly, the addition of 1 to 2 and then to 1000, gives a different result from addition of 1 to 1000 and then of 2. To avoid the more serious violation in the case of mixed sign numbers, two rules are followed: 1). add numbers of like magnitude together, 2). add small numbers together, 3). add the larger ones.

To assess software reliability, first one must understand the concepts of accuracy and precision. Accuracy and precision are different concepts. Following Altman et.al. (2004 pp. 20), accuracy refers to "the absolute or relative error of an approximation quantity," or the dissimilarity (or distance) between estimates and output. Referring to measurement, precision "refers to the degree of agreement among a set of measurements of the same quantity-the number of digits that are the same across repeated measurements,"(Altman et.al. 2004 pp. 15) or in another way "the number of bits in the significand (including the hidden bit)" (Overton 2001 pp. 14). For a single scalar value output, the log relative error (LRE) can be used as a measure of the number of correct digits in the output with respect to estimates.

$LRE = -log_{10}|\frac{output-e}{e}|$, where e is the benchmark estimate. In the test framework, the comparison to the benchmark needs to be set up in order to obtain the highest possible LRE value. An LRE value close to 0 indicates very inaccurate estimates. A less than 4 LRE may be problematic.

Significant digits of a number are defined as the first non-zero and succeeding digits. Doubling the number of zeros of the convergence tolerance (from 0.001 to 0.00001) can be used to increase the number of accurate digits i.e. the LRE. The StRD can be used to find the existence of a set of options producing better non-linear results than software defaults (McCullough 1999).

Chapter 6

Software Assessment

The content of the software assessment covers: estimator, algorithms, line search procedure, standard error estimation, tolerance level, and precision aspects.

These contents are compromised in the results given by software packages commands. The commands provided in software packages are listed in Tables 6.1-6.3.

SAS details are found in Table 6.1: in the "Proc Glimmix" command of SAS, Technique=NEWRAP denotes Newton-Raphson optimization combining a linear-search with ridging; Technique=CONGRA denotes conjugate-gradient optimization; Technique=QUANEW represents quasi-newton optimization (SAS Manual). This command fails many reliability tests and can be used as an alternative estimator.

In a model sentence of SAS commands, /Gconv denotes relative gradient criterion, and a value of 1E-8 are the default settings in the "Proc Logistic" and "Proc Surveylogistic" commands. In addition, in these two Procs "Absfconv", "Xconv", "Gconv" and "Fconv" denote absolute function convergence criterion, relative parameter criterion, gradient convergence criterion, and relative function convergence criterion respectively (SAS Manual).

In the "Proc Catmod" command, the log likelihood change can be specified by "Epsilon" in model sentence and a value of 1E-8 is the default setting.

In the model sentence of "Proc Genmod" command, the parameter estimate converge has its default value 1E-4 using "Converge=1E-4"; relative Hessian convergence can be set using "Convh=1E-4"; profile likelihood confidence interval is specified by "CIConv=1E-4".

Table 6.1: *Comparison of Software Commands in SAS*

SAS9.1						
Procedure	Logistic	CATMOD	GENMOD	SURLGT	QLIM	Glimmix
Estimator[a]	ML	ML;WLS	ML	ML	ML	Quasi-likelihood
Algorithm	FS	NR; Iterative	ridge-stabilized	FS;	Quasi-New	NR
	NR	Proportional Fitting	NR	NR	conjugate-gradient	QUA-NEW
					NR line-search w ridging	conjugate-gradient
Default	FS	NR	GEEs	FS	QUANEW	QUANEW
SE	EIM	OIM	OIM	Variance estimate, Linear	inverse Hessian matrix	2× observed inverse Hessian
				Approximate of Estimator		
Tolerance	gradt:1E-15		Para:1E-4	1E-15	F:2E-16	1E-8
Criterion	X:E-12		Hessian:1E-15	G:E-15	ABSG:E-5	
	likeli:1E-12	1E-8	1E-4	1E-12	1E-5	1E-8
Precision	double	double	double	double	VMSAlpha:	IEEE double precision
Default Modeling	P(Y=0)	P(Y=0)	P(Y=1)	P(Y=0)	P(Y=1)	P(Y=1)

Table 6.2: *Comparison of Software Commands in Limdep, Matlab, Shazam, R, and Minitab*

Software Name	Limdep9		Matlab7	Shazam10	R2.10.1	Minitab15
Command	Logit	Blogit	Logit	Logit	GLM	Binarylogit
Estimator	ML	ML	ML	ML	Quasi-L	ML
Algorithm	BHHH	BHHH	NR;Newton	DFP	FS; IWLS	IRLS
	BFGS, NEW	BFGS, NEW		BSGF		
Default Algorithm	BFGS	BFGS	NR	DFP		
Tolerance Criterion			1e-6	l:1E-3		20 iterations
SE			OIM	-H^{-1}OIM	IWLS	
Precision	double	double	double:15 Significant Digits	double	single:integers	double
Default	P(Y=1)	P(Y=1)	P(Y=1)	P(Y=1)	P(Y=1)	P(Y=1)

Table 6.3: *Comparison of Software Commands in Stata, Eviews, and SPSS*

Software	Stata10					Eviews3	SPSS17
command	Logit	GLM	Binreg	BHHH	BFGS	logit	ordinalreg
Estimator	ML	ML	MQL;ML	ML	ML	ML	ML
Algorithm	NR;BHHH	NR;	NR;BFGS	BHHH		quadratic climbing hill	
	DFP;BFGS	IRLS(MQL)	DFP;BHHH			Newton,BHHH	
Default Algorithm	NR	ML:NR	MQL Fish Scoring(IRLS)	BHHH		quadratic climbing hill	
Tolerance Criterion		Scaled gradient(1e-4)	ML(1e-6)	Scaled gradient(1e-5)			l:0,1E-5
	Scaled gradient(1e-4)		loglik(1e-7)	gradient(no check)			p:E-6,E-8
							s:E-8,E-10
SE	OIM	OIM	EIM	OPG	OIM	OPG	
Precision	double	double	double	double	double	double	double
Default	P(Y=1)	P(Y=1)	P(Y=1)	P(Y=1)	P(Y=1)	P(Y=1)	P(Y=0)

[a] Refer to List of Acronyms for all Tables

See manual for the sequence of the convergence levels. This command has been proved to be extremely disable in reliability tests. This GEE can be used as alternative estimate.

In table 6.2, R and Minitab use iteratively reweighted least squares (IWLS) to estimate the model.

In table 6.3, Stata commands use observed information matrix (OIM) in calculating standard errors. Newton-Raphson steps should use the Fisher scoring Hessian or expected information matrix (EIM) before switching to the observed information matrix (OIM). Some commands use outer product of the gradient (OPG) vectors to calculate standard error.

An algorithm may prove to work properly only for a subset of the data and model. Therefore, the existence of benchmark data is rationalized for the reason of consistent working for all data and models. To look for a “preferred combination” of options, reporting the estimation results with more significant digits is necessary.

The software packages have respective advantages: R and Matlab are good at graphic narrative. Minitab, Eviews, and SPSS have well build-in functional icons, which facilitate the implementing the functions. Stata and SAS are very multi-functional in employing various estimators in logistic modeling. Shazam and Limdep have very powerful nonlinear solvers and MLE.

Chapter 7

Data Properties and Benchmark Data Generation

According to the literature introduction on benchmark data suites, StRD includes five main suites: univariate summary statistics, analysis of variance, linear regression, Markov chain Monte Carlo, and nonlinear regression (Table 7.1). Within each suite, NIST has created problems with three different levels of difficulty: low, average, and high. NIST provides certified values computed with multiple precision in 500 digits of accuracy (Yalta 2007), rounding to 15 significant digits for linear regression and 11 significant digits for nonlinear least squares. In nonlinear regression data sets, starting values near and far from the solution are provided. The nonlinear problems are solved using quadruple precision (128 bits) on two public domain programs, using only double precision, with different algorithms and different implementations. The convergence criterion is the residual sum of squares with a 1E-36 tolerance level.

Low LREs indicate that a package might use a less robust solution method. To look for a "preferred combination," the NIST nonlinear benchmarks should be run using the first set of starting values and report the LREs of any solution. When no solution is produced, tinker with the preferred combination to find a solution from the first set of starting values by varying the tolerance, switching algorithms, or changing default derivatives from numerical to analytic. For the last trial, the second set of starting values can be used. In

Table 7.1: *Comparison of NIST Benchmark Datasets*

Properties	Univariate	ANOVA	Linear Regression	Nonlinear Regression
Number of Datasets	9	11	11	27
Lower Difficulty	6	4	2	8
Average Difficulty	2	4	2	11
Higher Difficulty	1	3	7	8
Certified Value Digits	15	15	15	11
Observation Number	3-5000	5-2001	3-82	6-250
Coefficient Number			1-11	2-9

assessing numerical accuracy, nonlinear benchmarks can be used to answer two questions in nonlinear estimation: whether the default options should be relied on? The combination of convergence tolerance, the method of solution, and the convergence criterion can change the solution. Are analytic derivative worth the effort? The answer is yes, because analytic derivatives are more accurate than their finite-difference approximations (McCullough 1998).

Forty years ago, Longley (1967) and Anscombe (1967) pioneered generating benchmark data sets for validating software in multiple linear regression procedures and standard deviation calculations. Algorithms were biased towards overstating numerical accuracy because benchmarks use integer values. Anscombe's benchmark shortcomings originate from using integer values with an exact binary representation, whose magnitudes can be stored without truncation in a computer. The exact benchmark omits the likelihood of existence for the inexact binary representation. Modifying the integers by dividing them by ten gives the inexact binary representation, which allows a study of their implications in algorithms. Alternative benchmarks in Simon and Lesage (1988) containing flexible control parameters provide a more realistic assessment of numerical accuracy with different ill-conditioned data sets.

Prior benchmarks (e.g. Lesage and Simon 1985) for testing the accuracy of matrix decomposition algorithms for solving the least squares problem in comparison with procedures used in many microcomputer statistical packages regression is pertinent to reliable results in the face of ill-conditioned data. The benchmark data set is made with an increasingly ill-conditioned manner that allows the continuum of the numerical accuracy to range from mild to extreme. Wampler (1980) uses the proposed matrix size and a small deviation from zeros, ϵ, on the (j,j)th place as the design regressor matrix. When the ϵ is zero, the $X'X$ matrix is singular, and when ϵ is close to zero, the $X'X$ matrix is near singular. When the relationship between the condition number and ϵ is formulated, the condition number increases with a decreasing ϵ (Wampler 1980).

The condition number is a concept defined in the numerical literature, as the product of matrix norms $\|X'X\|.\|(X'X)^{-1}\|$ of the linear regressor matrix (Sawitzki 1994), and regression problems are well-known to be potentially ill-conditioned (i.e. the $X'X$ matrix is close to singularity). The condition number is defined as "the ratio of the largest to the smallest singular values of X or the square root of the ratio of the largest to the smallest eigenvalues of $X'X$ (Lesage and Simon 1985 pp. 49). It measures to what extent the matrix is ill-conditioned: the larger a condition number is, the closer the matrix $X'X$ is to singularity. The behavior of an ill-conditioned situation results in inherent instability of the regression results, because any algorithm becomes affected by the ill-conditioned data. "The error from truncation or rounding of input data is the same order of magnitude as the estimated standard error of coefficients" (Sawitzki 1994 pp. 275).

Following Higham (2002 Secs. 1.5-1.6), conditioning is defined in the most general way as "the sensitivity of the model to perturbations of the data" (pp. 9). Condition numbers indicate the conditioning of a problem using a particular set of inputs. (McCullough 2004). The consequences of ill-conditioning may lead to a putative solution in a "flat" region of the parameter space: some parameter values change by large amounts without any change of the objective function. Therefore the verification of the maximum of the log-likelihood problem is difficult. Ill-conditioning leads to serious accumulative rounding error and inaccuracy in computation. McCullough and Vinod (2003) show that for an ill conditioned Hessian, the quadratic approximation fails to hold in at least one direction. The Wald inference will be unreliable for at lease one of the coefficients.

$$f_{Y|X}(Y_i|X_i;\beta)f_X(X_i;\theta_j) = f_{X|Y}(X_i;\theta_j)f_Y(Y_i;p) = f(Y_i, X_i;\varphi) \tag{7.1}$$

The data used in this study are generated using the inverse conditional distributions of X on Y with parameter θ to recover the joint distribution of X and Y depending on parameter φ, including β and θ, or conditional distribution of Y on X, depending on β. The parameters θ (including μ, σ, and ρ) used in generation are functions of Y. μ and σ are the means and standard errors of the distributions of covariates X_i/Y_i. ρ is the correlation among covariates (Bergtold et al. 2010). The above equation denotes the relationship described above for the simulation process (Scrucca and Weisberg 2004). It includes two steps: first generate the inverse conditional distribution of X conditional on Y, choosing parameters θ and P. Second, generate the conditioning binary Y with a Bernoulli distribution depending on parameter P.

$f_{Y|X}(Y_i|X_i;\beta)$ is the conditional distribution that is of interest. By re-expressing it with the equation after the first equality, the relationship on X can be obtained. $f_{X|Y}(X_i;\theta_j)$ is the inverse conditional distribution that can be simulated. $f_Y(Y_i;p)$ is the unconditional probability density of Y_i. The data generation process takes these parameter values as given and the β of the conditional mean can be recovered using these selected parameters. Instead of using X to cover Y, one can use $f_{X|Y}(X_i;\theta_j)$ to cover θ_j and p the probability of $Y = 1$, and therefore obtains $f_{Y|X}(Y_i|X_i;\beta)$.

Chapter 8

Experiments

To test the impact of different errors on the reliability of logistic regression estimation, experiments are constructed for multicollinearity, cut-off point, and functional form.

First, to test the effect of different degrees of multicollinearity and the impact on logistic regression reliability, this study performs tests by generating data with different numbers of observations (e.g. 200 to 20000) with up to 5 covariates, changing the amount of correlation between any two covariates with correlation from 0.75 to 0.995.

Next, the study examines the cut-off point. By changing the cut-off (i.e. P(Yi = 1)) from 0.01% to 50% with different numbers of observations (e.g. 32 to 20,000), the study assesses the effect of changing the cut-off ratio on reliability.

Finally, the study examines how functional forms affect reliability. To examine the effect of functional form has on the reliability of the logistic regression software packages, the forms of the predictor or index function in logistic model will be varied from linear in parameters, logarithm form, and quadratic.

The description of the datasets used to fulfill these experiments are presented in Table 8.1. The variance(X) denotes the variance of each covariate used in the simulation. The details of the performed experiments to fulfil testing the impact of convergence, algorithms, and starting values on the software reliability are in Table 8.2. All options evaluating from these three perspectives are tested throughout. The options are evaluated with each other as many combinations as possible in order to achieve a maximum LRE. The content on how default starting value of each software command is set will be described in the next section. The closest starting values are found by comparing an assumed closest certain value with respect to OLS and zero with the expected parameter values to determine which one is the closest.

Table 8.1: *Description of Benchmark Datasets*

Name	Base	Multi1	Multi2	Multi3	Multi4	Multi5	Multi6	Multi7	Mvar1	Mvar2	Mvar3	Mvar4	Cuto1	Cuto2	Cuto3	Cuto4
No. of Observation	200	500	500	500	1000	1000	1000	1000	300	300	1000	1000	50	32	5000	20000
Covariates	1	2	2	2	2	2	2	4	3	3	5	5	2	3	4	3
Linear terms	1	2	2	2	2	2	2	4	2	3	5	5	2	2	4	3
Quadratic terms	0	0	0	0	2	2	2	0	0	1	0	0	0	0	0	0
Interaction terms	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Logarithm forms	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0
Maximum Correlation	N/A	0.75	0.95	0.995	0.75	0.95	0.995	0.985	N/A	N/A	N/A	N/A	0.99	N/A	0.96	N/A
Variance(X)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	15	1	N/A	N/A	N/A	N/A
P(Y=1) [a]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0.05	0.15	0.0005	0.00015
Difficulty	low	low	average	high	low	average	high	high	low	average	average	average	average	average	high	average

Table 8.2: *Description of the Changing the Default Setting*

Name of Command	Changing Algorithms			Changing Convergence Level		Changing Starting Values			
SAS Logistic	Default	Fisher Scoring	Newton Raphson	Default	Xconv=1E-15	Default	Start1(0)	Start2(Closest)	Start3(OLS)
SAS Genmod	Default	Default	Default	Default	Convergence= 1E-15	Default	Start1(0)	Start2(Closest)	Start3(OLS)
SAS Qlim	Default	Default	Default	Default	Default	Default	Start1(0)	Start2(Closest)	Start3(OLS)
SAS Catmod	Default	Default	Default	Default	Epsilon=1E-15	Default	Start1(0)	Start2(Closest)	Start3(OLS)
SAS Surveylogistic	Default	Default	Default	Default	Gconv=1E-15	Default	Start1(0)	Start2(Closest)	Start3(OLS)
Limdep Logit	Default	BFGS	BHHH	Default	Tlg=1E-15	Default	Start1(0)	Start2(Closest)	Start3(OLS)
Limdep Blogit	Default	BFGS	BHHH	Default	Tlg=1E-15	Default	Start1(0)	Start2(Closest)	Start3(OLS)
Stata Binreg	Default	BFGS	BHHH	Default	Tol=1E-13	Default	Start1(0)	Start2(Closest)	Start3(OLS)
Stata GLM	Default	Default	Default	Default	Tol=1E-13	Default	Start1(0)	Start2(Closest)	Start3(OLS)
Shazam	Default	Default	Default	Default	Increase .00 till 1E-11	Default	Default	Default	Default
SPSS	Default	Default	Default	Default	Maximum	Default	Default	Default	Default
R GLM	Default	Default	Default	Default	Default	Default	Start1(0)	Start2(Closest)	Start3(OLS)
Minitab	Default	Default	Default	Default	Default	Default	Start1(0)	Start2(Closest)	Start3(OLS)
Eviews	Default	Newton Raphson	BHHH	Default	Increase .00 till 1E-7	Default	Default	Default	Default

[a] Available for every dataset.

Chapter 9

Results and Comments on Software Reliability

To test the impact of starting value on the estimation, four sets of starting values are used in estimating the results. The first is the default settings of the software, the second are starting from the zero vector (in most cases it is farther from the estimates), the third are relatively closer than zero starting values, and last are the ordinary least squares. The starting value can be specified in some logistic regression commands(e.g. Logistic, Genmod, and Qlim procs in SAS, Limdep, R, Minitab, Binreg in Stata, Eviews and their starting values are specified in next subsection) but in Catmod, Surveylogistic, and Glim procs in SAS, Matlab, Shazam, SPSS, GLM and Logit commands in Stata, only the default starting values can be used in estimation.

9.1 Dependence on Starting Values

For most software packages the default starting value for MLE estimation uses ordinary least square parameter estimates, while others use different methods (e.g. null vector). In Shazam, starting values for the slope coefficients of the index function or predictor are equal to zero, and the intercept is obtained by solving: $\frac{1}{1+exp(-\beta_1^0)} = S/N$, where N is the number of total observations; S is the number of observations with dependent variable $Y = 1$; S/N is the probability that $Y_t = 1$ for all observations.

In Limdep, starting values are either computed internally - OLS values are a frequent choice - or set to zero when the starting values do not matter (rarely).

During optimization, one may converge to a local maximum rather than a global one. One may avoid local maxima by running the estimation with different starting values or with a different maximization technique. The "START=" option in "Logit" and "Blogit" commands can be used to automatically perform a grid search to aid in the search for a global maximum.

In Eviews, a built-in algorithm using specific information about the objective function is used to determine the starting values.

In SAS, for "Proc Logistic", starting values can be specified in the logit regression by using "INEST= data set" to specify the input data set containing the initial estimates for all parameters. For the "Proc Qlim" command, "INIT" initializes the specification of starting values for each parameter estimates of a logistic regression model.

In nonlinear optimization, convergence can be expected only with fully identified parameters, adequate data, and starting values sufficiently close to solution estimates.

"Convergence and the rate of convergence may depend primarily on the choice of starting values for the estimates. This does not mean that a great deal of effort should be invested in choosing starting values" (SAS 9.1 online on starting value). It is suggested by SAS to try the default values first and if the estimation fails with these starting values, examine the model, data and re-estimate using reasonable starting values. It is usually not necessary that the starting values are very good, just that they are not very bad; choosing values that seem plausible for the model and data is helpful in obtaining convergence.

If the procedure fails to converge because it is unable to find a vector that improves the objective value, check the model and data used to ensure that all parameters are identified and data values are reasonably scaled. Then, re-estimate the model with different starting values. Consider using the Marquardt method if Gauss-Newton fails, because the Gauss-Newton method may fail if the Jacobian matrix is nearly singular or ill-conditioned. A nonlinear model may be well-identified and well-conditioned for parameter values close to the solution values but unidentified or numerically ill-conditioned for other parameter values. The choice of starting values may make a difference (SAS 9.1 online).

9.2 Software Reporting

To obtain estimation output with more significant digits, different software packages use different methods.

Table 9 shows, how to obtain more significant digits in software output, using commands in the different software packages examined.

The following commands can change the starting value (in Table 9.2), algorithms (in Table 9.3), and convergence level (in Table 9.4) for optimization estimation of the logit model.

SAS: “Proc logistic” has an option for changing the starting values by specifying the name in “INSET” with the imported and named vector of starting values in an individual dataset. Its default algorithm, fisher scoring, can be changed to Newton by setting “Technique=Newton”. In “Proc Surveylogistic”, the model sentence can be changed with “Fconv”: relative function convergence criterion, “Gconv”: 1E-8 as the default relative gradient convergence criterion, “Xconv”: relative parameter convergence criterion, “Absfconv”: absolute function convergence criterion to change consequent criteria. In “Proc Genmod”, the starting value can be changed by setting “initial= starting value vector”. In the model sentence, “converge”: 1E-4 as the default parameter estimates, “convh”: 1E-4 as the default relative hessian convergence, “CIconv”: 1E-4 as the default profile likelihood confidence interval. In “Proc Catmod”, the “Epsilon” has a default of 1E-8 in model sentence to denote the change in log likelihood.

Limdep: “Matrix ; Peek” ; Changing Starting value: “Start=0,0”; Changing algorithms: “Alg=BHHH”; “BFGS”: rank 2 update; “NEWTON”; “DFP”: rank 1 update ; “SteDes”. LIMDEP will choose either “Broyden” (BFGS) or “Newton” as the default. The value of

Table 9.1: *Comparison of Software Commands for Reporting More Significant Digits*

SAS	STATA	MATLAB	LIMDEP	SHAZAM	R	MINITAB	EVIEWS	SPSS
ODS Output	Format	Format	Peek	Format	Option(Digits)	Icon:Editor-format	Copy directly	Copy directly

Table 9.2: *Comparison of Software Commands in SAS, Limdep, R, Minitab, and Stata Capable of Changing Starting Values*

SAS-Genmod	SAS-Logistic	SAS-Qlim	Limdep (BFGS)	Limdep (BHHH)	R	Minitab	Stata-Binreg
Initial=values	INEST= vector name	INIT each defined	Start=values	Start=values	Start=c(vector)	Use icon	Init(vector name)

Table 9.3: *Comparison of Software Commands in SAS, Limdep, Eviews, and Stata Capable of Changing Algorithms*

Software	SAS-Logistic	SAS-Qlim	Limdep-Logit,Blogit	Eviews	Stata-Binreg
Command	Model Technique=	Technique=	Alg=	Change Icon	Tech()

Table 9.4: *Comparison of Software Commands in SAS, Limdep, Eviews, SPSS, Shazam, and Stata Capable of Changing Convergence Levels*

Command	SAS-Logistic	Catmod	Genmod	Slgt	Limdep-Logit,Blogit	Stata-ML	Shazam	SPSS	Eviews
Options	Fconv Gconv Xconv Absfconv	Epsilon	CLCONV convh converge	Fconv Gconv Xconv Absfconv	Tlg Tlb Tlf	Tol ltol gtol nrtol	conv	Log-l Para Singul	Icon
Defaults	1E-8 1E-8 1E-8 1E-8	1E-8	1E-4 1E-4 1E-4	1E-8 1E-8 1E-8 1E-8	1E-6 0 0	1E-6 1E-7 N/A 1E-5	1E-3	0 1E-6 1E-8	1E-3

convergence can be set in the model sentence, by using “Tlg” to denote value of change gradient with a default 1E-6, “Tlb” to denote value of parameter change with a default 0, “Tlf” to denote value of function change with a default 0. One can prevent any rule from being met by changing the tolerance to 0.0 by not including the “=value” part of the command. User supplied starting values are never necessary for the models estimated by LIMDEP. Iterative routines in LIMDEP normally require from 5 to 25 iterations. A large, complex problem can require many more. The default settings for maximum iterations are 25 for Newton’s method and 100 for Quasi-Newton. One can change the maximum iterations with “; Maxit = value” in any model command. Note, as well, that “; Maxit = 0” is a special command that requests computation of a LM statistic (as well as all other results) at the starting values (Limdep 9 Manual).

Moreover, the success of an algorithm depends on the data. Usually the default setting - “BFGS” and Quasi-Newton are the most reliable. In large datasets, “NEWTON” or “BHHH” may perform better, and “BFGS” or “DFP” if they fail to find the maximum. Newton’s method is best for logit model, because the objective function is globally concave. When the “BHHH” estimator is used to estimate the covariance matrix and fails to converge, Newton’s method will be used. Steepest descent is not suggested for use, because its convergence speed is slow and tends to fail (Limdep 9 Manual).

R-2.10.1: Unless the convergence criteria are made fairly tight, different algorithms will often provide with fifth or sixth significant digit answers. “options(digits = 15)” controls. Changing starting values can be realized by specifying “start=c(starting values)”.

Shazam, STATA, and Matlab: “Format” is used to specify the number of significant digits in output.

Stata10 “Binreg” can change the starting value and algorithms in estimation, using the “init(vector of starting value)” or “tech(name of the algorithm)”. Need to import and name a vector of starting values in the dataset. The tolerance level of change in log likelihood (“ltol”) has a default of 1E-7, change in gradient relative (“gtol”) to the coefficients does not have a value as default, and change in scaled gradient ("nrtol") has a default of 1E-5.

Minitab : One can follow the following steps to fulfill the testing: first use "File-Open worksheet- CSV file" to open the data, and then use "Stat-Regression-Binary logistic regression-Storage" to specify the information needed from the regression result; after that use "Editor - Format column - Numerical - Fixed decimal" - built-in icons, to obtain the number of precision digits one wishes to obtain. To specify starting value: first, one needs to change the icon in "Stat-Regression-Binary logistic regression"-Option and then import and name a vector of starting values in the dataset.

Eviews 3.1: After loading the data through "New-workfile- input time period", use "Procs-import" to input the data with relative names of variables and use "Genr" enter equation "z=1". Use icon "File-open-Program" to type in command "logit Y Z X", and run: in the equation: untitled, and workfile-untitled. Use the set icon "Stat-copy-unformatted" to reach the maximum precision. The starting points and algorithms can be changed using the estimate option icon-option, where one can change from the default "quadratic climbing hill" to "Newton" and "BHHH". The starting point can be changed from "Eviews provided" to "0.3×Eviews" and "zero". The convergence level can be changed by using different icons.

SPSS 17-"ordinal regression" default: "Maximum iterations": 100, "log-likelihood convergence": 0 (1E-5 is the maximum), "parameter convergence": 1E-6 (1E-8 is the maximum), and "Singularity tolerance": 1E-8 (1E-10 is the maximum). Threshold V_1=0 intercept has a wrong sign, because it measures the probability of $Y = 0$ by default. Double click each coefficient in the output provides numbers with more significant digits. Transform the variables using icon in transform - calculate variables.

9.3 Comments on Dependence on Algorithms

During estimation, algorithms can be changed from defaults to other algorithms (e.g. "Proc Logistic" in SAS, Limdep, Eviews, "Binreg" in Stata), but algorithms may not be changed in other software packages; such as "Catmod", "Genmod", "Surveylogistic", "Glimmix", and “Qlim” procs in SAS, “Logit” and “GLM” in Stata, Minitab, SPSS, R, Matlab, and Shazam.

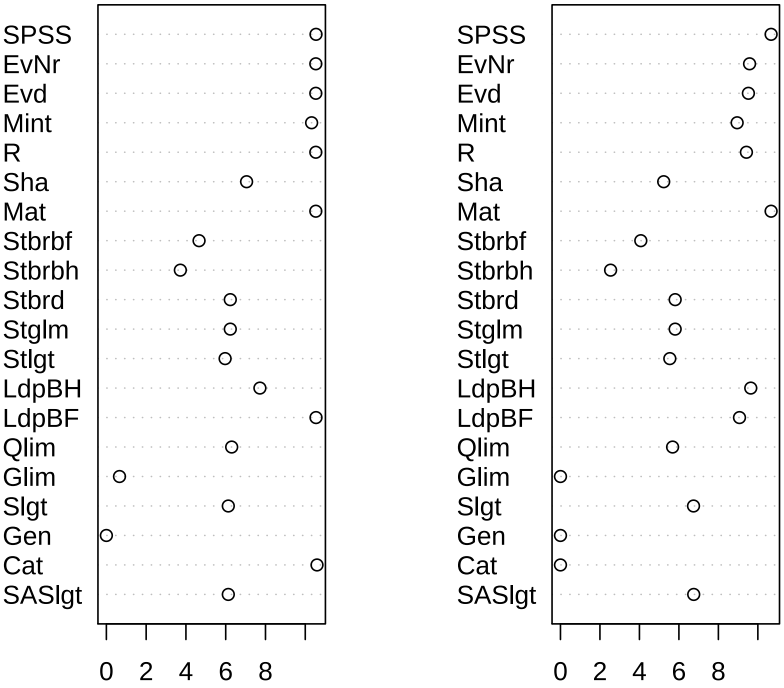
Based on the minimum coefficient LREs, the ranks of all commands for dataset1 (multi7) with 1000 observations and dataset2 (cuto3) with 5000 observations are in Figure 9.1. The legend for this figure is: SASlgt, Gen, Glim, Cat, Slgt, Qlim: SAS Logistic, Genmod, Glimmix, Catmod, Surveylogistic, Qlim; Stbrd, Stbrbh, Stbrbf, Stlgt, Stglm: Stata Binreg Default, BHHH, BFGS, Logit, and GLM; Sha: Shazam; Mint: Minitab; LdpBF, BH: Limdep BFGS, BHHH; Evd, EvNr: Eviews Default, NR; Mat: Matlab. Figure 1 summarizes the sensitivity of the commands depending on the change of number of observation. The left plot is for dataset multi7 with 1000 observations and the right plot is for dataset cuto3 with 5000 observations. As the number of observations increases, the performance of Eviews-Newton Raphson, Eviews-Default, Minitab, R, Shazam, Stata-Binreg(BFGS and BHHH), Limdep-Logit, Blogit (BFGS), SAS-Qlim Glimmix worsened. Whereas, SAS-Logistic Surveylogistic Limdep-Logit, Blogit (BHHH) improve their performance. SAS-Catmod can not estimate the larger dataset.

Identical algorithm provides the same results for the Limdep Logit and Blogit commands.

In Table 9.5, Stata, Logit GLM Binreg(default) can estimate coefficients of datasets multi1-7 up to the same reliable level but Binreg (BHHH and BFGS) does a worse job. In Table 9.8, however, for datasets mvar1-4 and cuto1-4, the coefficient LREs of Logit, GLM, Binreg (default) are greater than Binreg (BHHH), and Binreg (BFGS) with mvar3 as an exception. In Stata, Logit, GLM, Binreg(default), and Binreg (BFGS) can estimate standard errors reliably, but they are greater than standard error LREs provided by Binreg (BHHH) for all datasets except base. Log-likelihood LREs are relatively invariant to choice of algorithms.

In Table 9.14, for Eviews, BHHH has smaller coefficient and standard error LREs than the default setting and the Newton-Raphson option.

Figure 9.1: *The Rank of Software Commands-Minimum Coefficient LRE of Parameter Estimates for Multi7 (on the left with 1000 observations) and Cuto3 (on the right with 5000 observations) (Refer to List of Acronyms)*



9.4 Comments on Dependence on Observation Number

Tables 9.5, 9.8, 9.11, 9.15, 9.18, and 9.21 provide the LREs, on which figure 9.1 is based. SAS "Catmod" provides the highest coefficient LRE of 10.59 (12.34)[1] for multi7, but its performance depends on the number of observations of the dataset. For cuto3, that has a larger number of observations, it does not converge at all. The maximum LRE for SAS "Qlim" increases and minimum LRE decreases with an increasing number of observations, i.e. 6.30 (6.64) in multi7, and 5.68 (7.67) in cuto3. SAS "Logistic" and "Surveylogistic" LRE increases with an increasing number of observations, i.e. 6.13 (6.88) in multi7, and 6.75 (7.32) in cuto3. Limdep- Logit or Blogit "BFGS" LREs decreases with an increasing number of observations, i.e. 10.54 (11.61) in multi7, and 9.07 (9.47) in cuto3. Limdep "BHHH" LRE increases as the number of observations increases, i.e. 7.72 (8.08) in multi7, and 9.64 (10.92) in cuto3. Shazam LRE decreases with an increasing number of observations, i.e. 7.05 (7.91) in multi7, and 5.23 (5.71) in cuto3. Stata "GLM" maximum LRE increases and minimum LRE decreases with an increasing number of observations, i.e. 6.23 (6.46) in multi7, and 5.81 (7.19) in cuto3.

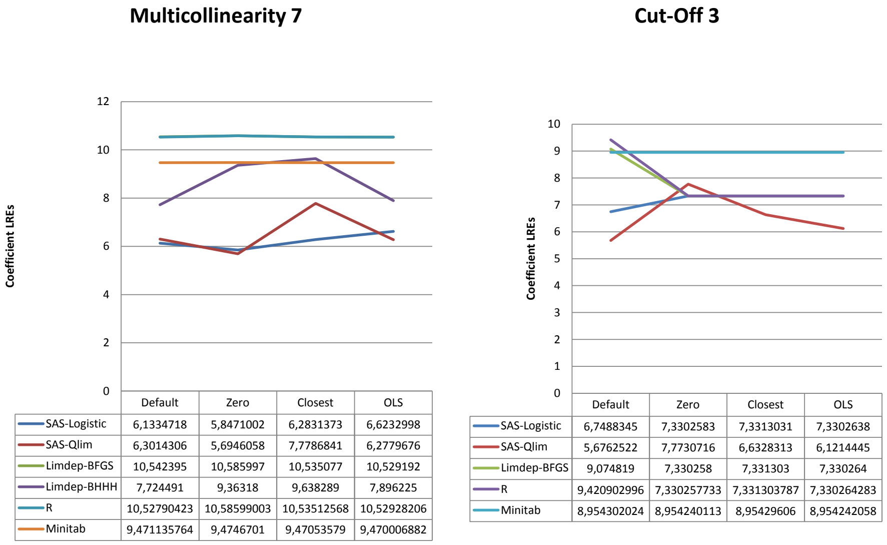
9.5 Comments on Dependence on Starting Values

Figures 9.2 and 9.3 describe how the starting values impact the coefficient and standard error LREs in Tables 9.7, 9.10, and 9.13.

SAS "Proc Logistic" is sensitive to starting values. Starting with the closest and OLS points provide larger LREs, while default starting values or zero provide smaller LREs. For multi7, Table 9.7 shows coefficient minimum LRE changes from closest (6.3) and OLS (6.6) to default (6.1) and zero (5.8). Standard error minimum LRE changes from closest (6.0) and OLS (6.4) to default (5.9) and zero (5.6). For cuto3, Table 9.17 shows the coefficient

[1] The number in parentheses besides the minimum LRE is the maximum LRE. This convention is used throughout the remainder of the dissertation.

Figure 9.2: *Comparison of the minimum Coefficient LREs for Different Starting Values for datasets Multi7 and Cuto3 - Refer to List of Acronyms*



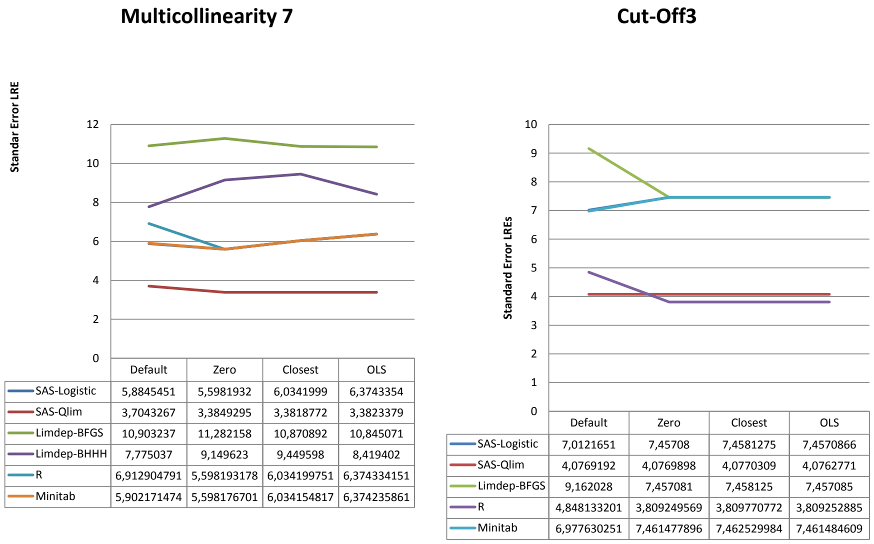
Multicollinearity 7

	Default	Zero	Closest	OLS
SAS-Logistic	6,1334718	5,8471002	6,2831373	6,6232998
SAS-Qlim	6,3014306	5,6946058	7,7786841	6,2779676
Limdep-BFGS	10,542395	10,585997	10,535077	10,529192
Limdep-BHHH	7,724491	9,36318	9,638289	7,896225
R	10,52790423	10,58599003	10,53512568	10,52928206
Minitab	9,471135764	9,4746701	9,47053579	9,470006882

Cut-Off 3

	Default	Zero	Closest	OLS
SAS-Logistic	6,7488345	7,3302583	7,3313031	7,3302638
SAS-Qlim	5,6762522	7,7730716	6,6328313	6,1214445
Limdep-BFGS	9,074819	7,330258	7,331303	7,330264
R	9,420902996	7,330257733	7,331303787	7,330264283
Minitab	8,954302024	8,954240113	8,95429606	8,954242058

Figure 9.3: *Comparison of the minimum Standard Error LREs for Different Starting Values for datasets Multi7 and Cuto3 - Refer to List of Acronyms*



Multicollinearity 7

	Default	Zero	Closest	OLS
SAS-Logistic	5,8845451	5,5981932	6,0341999	6,3743354
SAS-Qlim	3,7043267	3,3849295	3,3818772	3,3823379
Limdep-BFGS	10,903237	11,282158	10,870892	10,845071
Limdep-BHHH	7,775037	9,149623	9,449598	8,419402
R	6,912904791	5,598193178	6,034199751	6,374334151
Minitab	5,902171474	5,598176701	6,034154817	6,374235861

Cut-Off3

	Default	Zero	Closest	OLS
SAS-Logistic	7,0121651	7,45708	7,4581275	7,4570866
SAS-Qlim	4,0769192	4,0769898	4,0770309	4,0762771
Limdep-BFGS	9,162028	7,457081	7,458125	7,457085
R	4,848133201	3,809249569	3,809770772	3,809252885
Minitab	6,977630251	7,461477896	7,462529984	7,461484609

minimum LRE changes from zero (7.3), closest (7.3), and OLS (7.3) to default (6.7). Standard error minimum LRE changes from zero (7.5), closest (7.5), and OLS (7.5) to default (7.0). Log-likelihood LRE does not seem to be as sensitive as coefficient and standard error minimum LREs to changes in starting values.

SAS "Proc Qlim" is sensitive to changing starting values and starting with the null and closest starting values providing the highest LREs, but this depends on the dataset. For multi7, zero provides the smallest LRE (5.7) while the closest starting values provide the largest LREs (7.8), with default LRE and OLS LRE values in-between. However, for cuto3, starting from zero provides the largest LRE (7.8), but default LRE is the smallest with zero and OLS in-between.

Limdep-BHHH provides the most variation in LREs depending on the starting values. Table 9.10 shows, for multi7, coefficient minimum LREs change from 7.7 (default) and 7.9 (OLS) to 9.6 (closest) and 9.3 (zero), and standard error minimum LREs change from 7.8 (default) and 8.4 (OLS) to 9.4 (closest) and 9.1 (zero). As Table 9.20 shows, for cuto3, changing from the default starting values to other values shows a problem of convergence to the maximum of the log-likelihood function, and in the latter three cases estimators fail to provide a non-flat log-likelihood function. However, for cuto1, changing from default starting values to other three values converges to the maximum of a non-flat log-likelihood function. The log-likelihood LRE does not seem as sensitive as coefficient and standard error minimum LRE to changing starting values. In Limdep, for the BHHH algorithm, minimum coefficient LREs of starting from zero estimation are the smallest for dataset multi1. In the BHHH algorithm, no consistently best starting value can be found for all coefficients, but it never reaches the smallest LRE in starting with the closest points in all multicollinearity datasets and only reaches the smallest coefficient LRE in cut-off4 of multivariate and cut-off datasets. For the BFGS algorithm, the smallest minimum coefficient and standard error LREs locates amoung starting from default, zero, closest and OLS sets depends on which dataset is used, but all multicollinearity, multivariate, and cut-off (except cuto2) datasets end up with the smallest coefficient and standard error LREs when starting from OLS and zero starting points.

When using different starting values during estimation, the results in Stata10 Binreg are invariant to the change of starting values in Tables 9.9 and 9.19.

Minitab LREs are very insensitive to changing starting values. In Table 9.13 and Table 9.23 some datasets are insensitive to changing starting values, e.g. the log-likelihood LRE of most datasets, coefficient LRE of base, mvar1, 3, cuto1, 2, 3, multi1, 4, 6, and 7, and standard error LRE of base, mvar3. Other datasets are sensitive to changing starting values. In Minitab, starting from default, zero, and a closer set provide higher and identical coefficient LREs than starting from OLS. However, starting from a closer starting point provides the highest standard error LREs, followed by the default then zero with OLS performing the worst. An interesting finding is that the standard error LREs are differently ranked from coefficient LREs. For different datasets, the LREs for different starting values are in different magnitude ranges.

In Tables 9.13 and 9.23, the dependence of R on starting values changes with each test case. For multi7, the coefficient LREs do not depend on starting values, but for cuto3, the default starting values provide the largest minimum coefficient LRE than the other three independent starting values. R with default starting values is the best in providing the largest coefficient and standard error minimum LRE for base, multi1-6, mvar4, cuto3, and closer or OLS starting values improve the LREs compared to starting from zero.

In Table 9.14, when Eviews starts from zero it improves coefficient LREs over the default setting on multi6 and multi3, but decreases coefficient LREs when the Newton-Raphson algorithm is used with starting values (with multi5 as an exception). Changing starting values to zero impacts the standard error LREs, but not always in the same direction as the coefficient LREs.

9.6 Comments on Dependence on Convergence Level

In SAS, in Tables 9.6 and 9.16 decreasing the convergence level for relative parameter convergence "Xconv" or absolute function value "ABSFconv" improves the LREs of the results starting at OLS to the maximum extreme. Changing the convergence level of gradient "Gconv" enhances the LREs of coefficients as well, but with less magnitude. In both changes, LREs of the standard errors in "Proc Catmod" can also be enhanced by changing the log-likelihood "Epsilon". "Proc Catmod" is sensitive to the starting values, with the default and OLS providing the largest LREs compared to starting from zero and closer points.

In Tables 9.9 and 9.19, for Limdep, changing convergence level can improve the LREs for the BFGS and BHHH algorithms.

In Tables 9.9 and 9.19, for Stata, changing convergence level can improve the LREs in Binreg for all datasets and GLM algorithms only for some datasets (e.g. GLM for multi2 and multi4), but not to the maximum level LRE equal to 11.

In Tables 9.14 and 9.24, for Eviews, BHHH estimates smaller coefficient and standard error LREs than the default setting and Newton-Raphson algorithm. Changing convergence levels improves coefficient and standard error LREs from the default setting significantly for some coefficients, but influences all standard error LREs except cuto1. It does not improve the log-likelihood LRE, when changing the convergence level. FOR Newton-Raphson, changing the convergence level improves coefficient and standard error LREs, but does not improve log-likelihood LREs. For BHHH, changing the convergence level improves coefficient LREs, but does not improve standard error and log-likelihood LREs.

In Tables 9.12 and 9.22, for Shazam, changing the convergence level to 1E-11 improves coefficient and standard error LREs to 11, except coefficient LREs of cuto1. The exact convergence level of obtaining LRE=11 depends on the dataset.

In Tables 9.12 and 9.22, for SPSS, changing convergence levels to the maximum does not improve coefficient and standard error LREs.

The impact from the number of observations, covariates, and degree of multicollinearity are as follows.

9.7 Tests Based on Datasets Multi1-7

SPSS provides high LREs (10 to 11) for the datasets with a high degree of multicollinearity and/or multiple functional covariates (multi 3 to 7). However, it provides lower LREs (less than 10) for the datasets with less observations and relatively low degree of multicollinearity (multi1 and multi2). SPSS provides very close values between standard error LREs and coefficient LREs (Table 9.11).

Matlab's performance is very steady for all datasets and provides consistently high (more than 10) LREs for coefficient and standard error except multi1. The coefficient and standard error LRE magnitudes are very close (Table 9.11).

In R, the coefficient LREs are very high (above 10) for almost all multicollinearity datasets except multi5. R can not provide an equivalently high LRE for standard errors, which can only achieve about 6 or 7 as the maximum for multicollinearity datasets and 4 to 5 for multivariate datasets in Table 29. Because the threshold is 4 as problematic, the inaccuracy is tolerable. Because all standard error LREs are more than 4, and they dont indicate any problematic information (Table 9.11).

Shazam provides the most variation in coefficient LREs from high (above 10) multi2, and multi1, to relatively low (above 5) for other datasets (e.g. multi4). For all parameters in one model, the coefficient LREs are very closely aligned. The standard error LREs are only about half of the coefficient LREs (Table 9.11).

In Minitab, the deviation in LREs shows up in the different parameters in one model and between coefficients and standard errors, but not among different datasets. For the multi7 dataset, the coefficient LREs range from 9 to 11, which has a difference up to 2. In some datasets the difference between coefficient and standard error LREs are minor, such as in multi1, where the difference is only 1, but in other dataset with high degree of multicollinearity the difference is very significant up to 5 or 6 in multi3, and 4 in multi6.

Table 9.5: *Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in SAS for Testing Multicollinearity*

Proc	SAS-Logistic(Fisher)	Logistic(NR)	Catmod	Genmod	Surveylogistic	Glimmix	Qlim
Parameter Estimates							
Base	5.54(5.67)	5.54(6.67)	10.86(10.97)	1.12(1.36)	5.54(5.67)	1.12(1.36)	6.43(7.85)
Multi1	3.80(3.84)	3.80(3.84)	7.51(7.61)	0.62(2.15)	3.80(3.84)	1.62(2.15)	8.05(9.50)
Multi2	4.60(4.62)	4.60(4.62)	8.98(9.00)	1.38(1.46)	4.60(4.62)	1.38(1.46)	10.31(10.63)
Multi3	5.27(5.29)	5.27(5.29)	9.96(10.08)	1.33(1.39)	5.27(5.29)	1.33(1.39)	9.53(10.05)
Multi4	4.42(5.42)	4.42(5.42)	8.71(9.75)	0.40(1.08)	4.42(5.42)	0.40(1.08)	7.87(9.43)
Multi5	3.71(3.99)	3.71(3.99)	7.20(7.46)	0.73(0.99)	3.71(3.99)	0.73(0.99)	8.07(8.26)
Multi6	5.66(5.82)	5.66(5.82)	0(0.14)	0.71(0.80)	5.66(5.82)	0.71(0.80)	8.11(8.49)
Multi7	6.13(6.88)	6.13(6.88)	10.59(12.34)	0(0)	6.13(6.88)	0.66(1.35)	6.30(6.64)
Standard Error Estimates							
Base	6.23(6.64)	6.23(6.64)	5.79(6.18)	0(0)	1.37(1.84)	0.89(2.42)	7.99(8.40)
Multi1	4.32(4.52)	4.32(4.52)	4.14(4.34)	0(0)	1.41(1.44)	1.05(1.42)	5.15(6.07)
Multi2	4.75(4.83)	4.75(4.83)	4.58(4.66)	0(0)	1.37(1.96)	1.08(1.46)	4.68(4.86)
Multi3	5.04(5.04)	5.04(5.04)	4.89(4.89)	0(0)	1.21(1.24)	0.96(1.09)	3.08(3.08)
Multi4	4.91(5.54)	4.91(5.54)	4.70(5.32)	0(0)	1.14(1.84)	0.32(1.74)	4.76(5.48)
Multi5	4.02(4.34)	4.02(4.34)	3.86(4.18)	0(0)	1.13(1.67)	0.34(0.88)	4.63(5.56)
Multi6	6.01(6.19)	6.01(6.19)	0.00(0.18)	0(0)	1.37(2.44)	0.49(0.87)	2.44(3.75)
Multi7	5.88(6.38)	5.88(6.38)	5.60(6.09)	0.66(1.36)	0.47(1.92)	0.09(0.15)	3.38(5.29)
Log-likelihood Estimates							
Base	10.54	10.54	10.53	0.19	10.54	1.05	10.53
Multi1	8.43	8.43	10.87	NS	8.43	0	10.87
Multi2	9.77	9.77	10.72	0	9.77	0	10.72
Multi3	11.05	11.05	11.12	0	11.05	0	11.12
Multi4	10.32	10.32	11.0	0.20	10.32	0	11.18
Multi5	8.58	8.58	11.14	0.17	8.58	0	11.14
Multi6	11.96	11.96	1.16	0.05	11.96	0	12.05
Multi7	11.29	11.29	11.32	0	11.29	0	11.31

[a]

[a] NS:The package doesn't converge in estimation. Both minimum and maximum LREs are provided as a glance on the range of parameter and standard error LRE case in default setting for each software package. A large minimum LRE is usually associated with a large maximum LRE, but it is not always the truth. The maximum LRE provides the highest level of reliability that the estimation can achieve. An LRE less than 4 may be problematic.

Table 9.6: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in SAS for Testing Multicollinearity*

Proc	LGT(Fisher1)	LGT(Fisher2)	LGT(NR1)	LGT(NR2)	GEN1	GEN2	QLIM1	QLIM2	FISHER1X	FISHER2X	NR1X	NR2X	CAT,EP	SLGT,G
Parameter Estimates														
Base	5.1	6.6	5.1	6.6	1.1	1.1	6.9	6.4	11.0	11.0	11.0	11.0	11.0	11.0
Multi1	7.5	4.4	7.5	4.4	NS	NS	5.0	8.2	10.4	10.4	10.4	10.4	10.4	7.9
Multi2	4.4	5.2	4.4	5.2	NS	NS	10.0	10.3	11.0	11.0	11.0	11.0	9.0	9.3
Multi3	5.1	5.8	5.1	5.8	NS	NS	6.9	9.5	10.8	10.8	10.8	10.8	10.8	10.2
Multi4	4.2	5.3	4.2	5.3	0.4*	0.4*	7.6	8.4	10.4	10.4	10.4	10.4	8.8	9.1
Multi5	7.2	4.4	7.2	4.4	0.7	0.7	8.2	8.1	10.5	10.5	10.5	10.5	7.2	7.5
Multi6	5.5	6.9	5.5	6.9	0.7	0.7	8.7	8.6	10.8	10.8	10.8	10.8	0	10.9
Multi7	5.8	6.6	5.8	6.6	NS	NS	7.3	6.3	10.5	10.5	10.5	10.5	10.5	10.5
Standard Error Estimates														
Base	5.8	7.3	5.8	7.3	0	0	7.7	8.6	11.0	11.0	11.0	11.0	11.0	1.4
Multi1	8.0	4.9	8.0	5.0	NS	NS	6.4	6.7	11.0	11.0	11.0	11.0	8.0	1.4
Multi2	4.6	5.4	4.6	5.4	NS	NS	7.2	7.3	10.9	10.9	10.9	10.9	9.1	1.4
Multi3	4.9	5.5	4.9	5.5	NS	NS	5.7	5.7	11.0	11.0	11.0	11.0	9.8	1.2
Multi4	4.7	5.8	4.7	5.8	0	0	7.3	7.1	10.7	10.7	10.7	10.7	9.2	1.1
Multi5	7.5	4.7	7.5	4.7	0	0	6.6	6.6	10.7	10.7	10.7	10.7	7.5	1.1
Multi6	5.8	7.2	5.8	7.2	0	0	4.8	5.2	10.0	9.9	9.9	9.8	0	1.4
Multi7	5.6	6.4	5.6	6.4	NS	NS	5.9	5.9	10.9	10.9	10.9	10.9	11.0	0.5
Log-likelihood Estimates														
Base	10.6	10.6	10.6	10.6	0.2	0.2	10.6	10.6	10.6	10.6	10.6	10.6	10.6	10.6
Multi1	10.9	9.7	10.9	9.7	NS	NS	10.9	10.9	10.9	10.9	10.9	10.9	10.9	10.9
Multi2	9.5	10.6	9.5	10.6	NS	NS	10.7	10.7	10.7	10.7	10.7	10.7	10.7	10.7
Multi3	10.6	11.0	10.6	11.0	NS	NS	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Multi4	9.9	11.0	9.9	11.0	0.2	0.2	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Multi5	11.0	10.1	11.0	10.1	0.2	0.2	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Multi6	11.0	11.0	11.0	10.1	0.1	0.1	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Multi7	11.0	11.0	11.0	11.0	NS	NS	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0

[a]

[a] NS:The package doesn't converge in estimation. *The relative Hessian convergence criterion of 1.420694E-13 is greater than the limit of 1E-15. The convergence is questionable. When only two sets of starting values show up in the table, command or algorithm with 1 denotes zero starting point and command or algorithm with 2 denotes OLS starting point. The same rule holds for the following tables.

Table 9.7: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in SAS for Testing Multicollinearity*

Proc	LGT(def)	LGT(1)	LGT(2)	LGT(3)	Genmod(def)	Genmod(1)	Genmod(2)	Genmod(3)	QLIM(def)	QLIM(1)	QLIM(2)	QLIM(3)
Parameter Estimates												
Base	5.5	5.1	4.5	6.6	1.1	1.1	1.1	1.1	6.4	6.9	8.3	6.4
Multi1	3.8	7.5	3.8	4.4	0.6	NS	NS	0.6	8.0	5.0	7.7	8.2
Multi2	4.6	4.4	5.2	4.7	1.4	NS	NS	NS	10.3	10.0	7.3	10.3
Multi3	5.3	5.1	5.8	5.3	1.3	NS	NS	NS	9.5	6.9	9.5	10.3
Multi4	4.4	4.2	4.5	5.3	0.4	0.4	0.4	0.4	7.9	7.6	7.2	7.7
Multi5	3.7	7.2	4.3	4.4	0.7	0.7	0.7	0.7	8.1	8.2	9.0	8.1
Multi6	5.7	5.5	6.7	6.9	0.7	0.7	0.7	0.7	8.1	8.4	8.6	8.5
Multi7	6.1	5.8	6.3	6.6	0.7	NS	NS	NS	6.3	5.7	7.8	6.3
Standard Error Estimates												
Base	6.2	5.8	5.4	7.3	0	0	0	0	8.2	7.7	8.3	8.6
Multi1	4.3	8.0	4.4	4.9	0	NS	NS	0	7.0	6.4	6.6	6.6
Multi2	4.7	4.6	5.4	4.9	0	NS	NS	NS	4.7	4.7	4.7	4.7
Multi3	5.0	4.9	5.5	5.0	0	NS	NS	NS	3.1	3.1	3.1	3.1
Multi4	4.9	4.7	5.0	5.8	0	0	0	0	4.8	4.8	4.8	4.8
Multi5	4.0	7.5	4.6	4.7	0	0	0	0	4.6	4.6	4.6	4.6
Multi6	6.0	5.8	7.1	7.2	0	0	0	0	2.4	2.4	2.4	2.4
Multi7	5.9	5.6	6.0	6.4	0	NS	NS	NS	3.7	3.4	3.4	3.4
Log-likelihood Estimates												
Base	10.6	10.6	10.6	10.6	0.2	0.2	0.2	0.2	10.6	10.6	10.6	10.6
Multi1	8.4	10.9	8.5	9.7	0.1	NS	NS	0.1	10.9	10.9	10.9	10.9
Multi2	9.8	9.5	10.6	10.0	0	NS	NS	NS	10.7	10.7	10.7	10.7
Multi3	11.0	10.6	11.0	11.0	0	NS	NS	NS	11.0	11.0	11.0	11.0
Multi4	10.3	9.9	10.5	11.0	0.2	0.2	0.2	0.2	11.0	11.0	11.0	11.0
Multi5	8.6	11.0	9.7	10.0	0.2	0.2	0.2	0.2	11.0	11.0	11.0	11.0
Multi6	11.0	11.0	11.0	11.0	0	0	0	0	11.0	11.0	11.0	11.0
Multi7	11.0	11.0	11.0	11.0	0	NS	NS	NS	11.0	11.0	11.0	11.0

[a]

[a] NS:The package doesn't converge in estimation.

Table 9.8: *Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in Limdep and Stata for Testing Multicollinearity*

Proc	Limdep(BFGS)	Limdep(BHHH)	Stata-Logit	Stata-GLM	Stata-Binreg	Stata-Binreg(BHHH)	Stata-Binreg(BFGS)
Parameter Estimates							
Base	11.71(11.85)	6.65(6.99)	7.65(8.28)	7.65(8.28)	7.45(7.90)	7.65(8.28)	6.48(6.53)
Multi1	7.88(7.97)	10.39(12.13)	5.59(7.77)	5.59(7.77)	5.59(7.77)	3.64(4.80)	2.72(4.19)
Multi2	9.30(9.32)	8.55(8.65)	7.20(7.52)	7.04(7.24)	7.21(7.53)	4.55(4.64)	5.56(5.80)
Multi3	9.96(10.08)	7.54(7.57)	7.00(7.21)	6.78(6.89)	7.06(7.31)	4.13(4.21)	6.21(6.23)
Multi4	9.13(10.25)	8.72(8.90)	6.79(7.58)	6.49(7.78)	6.79(7.57)	3.65(5.76)	3.65(5.79)
Multi5	7.52(7.78)	7.05(8.67)	6.21(7.03)	6.21(7.04)	6.21(7.02)	4.02(4.61)	4.16(6.06)
Multi6	10.89(11.81)	8.54(8.86)	5.44(5.88)	5.49(5.96)	5.44(5.88)	4.23(4.60)	4.39(5.80)
Multi7	10.54(11.61)	7.72(8.08)	5.97(6.15)	6.23(6.46)	6.23(6.46)	3.72(4.26)	4.66(5.38)
Standard Error Estimates							
Base	12.42(12.82)	7.58(7.91)	6.23(6.58)	7.75(8.34)	6.57(6.91)	7.75(8.34)	7.03(7.75)
Multi1	8.41(8.60)	11.12(11.44)	7.88(9.73)	7.77(8.64)	6.78(6.95)	1.36(1.50)	4.72(4.88)
Multi2	9.46(9.54)	8.80(8.90)	4.75(4.83)	7.21(8.83)	6.07(6.12)	1.31(1.71)	5.95(6.10)
Multi3	9.76(9.80)	7.32(7.33)	5.04(5.04)	6.51(6.55)	5.94(5.95)	1.10(1.14)	5.96(5.97)
Multi4	9.64(10.36)	9.00(9.48)	4.90(5.54)	7.17(8.62)	5.92(6.60)	1.20(2.04)	4.98(6.02)
Multi5	7.83(8.14)	7.89(8.18)	6.71(7.07)	6.70(7.04)	4.75(5.07)	1.19(2.11)	5.31(6.95)
Multi6	9.55(9.74)	9.33(10.17)	6.35(7.46)	6.31(6.48)	6.31(6.49)	1.56(2.93)	5.90(6.39)
Multi7	10.90(11.55)	7.78(8.84)	6.03(6.28)	5.09(6.73)	5.08(6.64)	0.23(2.71)	5.04(6.81)
Log-likelihood Estimates							
Base	10.53	10.57	7.90	7.90	NA	7.90	7.90
Multi1	10.87	10.87	8.54	8.54	NA	8.53	8.40
Multi2	10.72	10.72	8.18	8.17	NA	8.18	8.17
Multi3	11.12	11.12	6.76	6.76	NA	6.72	6.76
Multi4	11.18	11.18	7.72	7.72	NA	7.73	7.72
Multi5	11.14	11.14	7.11	7.11	NA	7.11	7.11
Multi6	12.05	12.05	6.86	6.86	NA	6.86	6.86
Multi7	11.32	11.32	6.41	6.33	NA	6.33	6.28

[a]

[a] NA:not applicable.

Table 9.9: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in Limdep and Stata for Testing Multicollinearity*

Proc	Limdep-BF1	-BF2	-BH1	-BH2	-BF1-15	-BF2-15	-BH1-15	-BH2-15	Stata-Binreg(Bh1)	-Binreg(Bh2)	-Binreg(Bf1)	-Binreg(Bf2)	-Binreg(default)-1
Parameter Estimates													
Base	10.9	6.6	8.0	6.6	11.0	11.0	8.0	11.0	4.9	4.9	6.5	6.5	7.5
Multi1	7.5	9.1	7.0	10.4	10.4	10.4	7.1	10.4	3.6	3.6	2.7	2.7	5.6
Multi2	9.0	10.5	10.6	8.9	11.0	11.0	11.0	11.0	4.5	4.5	5.6	5.6	7.2
Multi3	10.0	10.7	9.3	8.3	10.8	10.8	9.3	8.3	4.1	4.1	6.2	6.2	7.1
Multi4	8.7	10.4	8.1	8.5	10.4	10.4	10.4	10.4	3.6	3.6	3.7	3.7	6.8
Multi5	7.2	9.0	7.8	8.2	10.5	10.5	10.5	10.5**	4.0	4.0	4.2	4.2	6.2
Multi6	10.8	6.9	7.9	9.9	10.8	10.8	10.8	10.7	4.2	4.2	4.4	4.4	5.4
Multi7	10.6	10.5	9.4	7.9	10.5	10.5	10.5	10.5	3.7	3.7	4.7	4.7	6.2
Standard Error Estimates													
Base	11.0	7.3	9.0	7.8	11.0	11.0	9.0**	11.0	1.4	1.4	7.1	7.1	7.7
Multi1	8.0	9.7	8.1	11.0	11.0	11.0	8.1**	11.0**	1.3	1.3	4.7	4.7	7.8
Multi2	9.1	10.5	10.6	9.0	10.9	10.9	10.6**	10.9	1.3	1.3	6.0	6.0	7.3
Multi3	9.8	10.8	9.0	8.1	11.0	11.0	9.0**	8.1**	1.1	1.1	6.0	6.0	6.8
Multi4	9.2	10.6	9.7	9.4	10.7	10.7	10.7	10.7	1.2	1.2	5.0	5.0	7.3
Multi5	7.5	9.3	8.1	9.5	10.7	10.7	10.7	10.7**	1.2	1.2	5.3	5.3	6.7
Multi6	10.0	7.2	8.8	9.9	9.8	10.3	10.1	10.3	1.6	1.6	5.9	5.9	6.2
Multi7	11.0	10.8	9.1	8.4	10.8	10.9	10.9	10.9	0.2	0.2	5.0	5.0	5.1
Log-likelihood Estimates													
Base	10.6	10.6	10.6	10.6	10.6	10.6	10.6**	10.6**	7.9	7.9	7.9	7.9	7.9
Multi1	10.9	10.9	10.9	10.9	10.9	10.9	10.9**	10.9**	8.5	8.5	8.4	8.4	8.5
Multi2	10.7	10.7	10.7	10.7	10.7	10.7	10.7**	10.7	8.2	8.2	8.2	8.2	8.2
Multi3	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0**	6.7	6.7	6.8	6.8	6.8
Multi4	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	7.7	7.7	7.7	7.7	7.7
Multi5	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0**	7.1	7.1	7.1	7.1	7.1
Multi6	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	6.9	6.9	6.9	6.9	6.9
Multi7	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	6.3	6.3	6.3	6.3	6.3

[a]

[a] **The log likelihood is flat at the current estimates. Indices following software algorithms -15 and -13: Convergence levels are 1E-15 and 1E-13.

Table 9.10: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in LIMDEP for Testing Multicollinearity*

Proc	Limdep-BFGS(def)	BFGS(1)	BFGS(2)	BFGS(3)	BHHH(def)	BHHH(1)	BHHH(2)	BHHH(3)
Parameter Estimates								
Base	11.0	10.9	10.0	6.6	6.7	8.0	11.0	6.6
Multi1	7.9	7.5	7.9	9.1	10.4	7.0	10.2	10.4
Multi2	9.3	9.0	9.6	10.5	8.5	10.6	10.2	8.9
Multi3	10.2	10.0	10.7	10.2	7.5	9.3	8.3	9.2
Multi4	9.1	8.7	9.4	10.4	8.7	8.1	8.6	8.5
Multi5	7.5	7.2	8.6	9.0	7.1	8.1	7.6	8.2
Multi6	10.9	10.8	10.8	6.9	8.5	7.9	9.8	9.9
Multi7	10.5	10.5	10.5	10.5	7.7	9.4	9.6	7.9
Standard Error Estimates								
Base	11.0	11.0	10.9	7.3	7.6	9.0	11.0	7.8
Multi1	8.4	8.0	8.5	9.7	11.0	8.1	11.0	11.0
Multi2	9.5	9.1	9.7	11.0	8.8	10.6	10.3	9.0
Multi3	10.0	9.8	10.8	10.0	7.3	9.0	8.1	9.0
Multi4	9.6	9.2	9.9	10.6	9.0	9.7	10.3	9.4
Multi5	7.8	7.5	8.9	9.3	7.9	8.1	8.4	9.5
Multi6	9.6	10.0	10.4	7.2	9.3	8.8	10.0	9.9
Multi7	10.9	11.0	10.9	10.8	7.8	9.1	9.4	8.4
Log-likelihood Estimates								
Base	10.6	10.6	10.6	10.6	10.6	10.6	10.6	10.6
Multi1	10.9	10.9	10.9	10.9	10.9	10.9	10.9	10.9
Multi2	10.7	10.7	10.7	10.7	10.7	10.7	10.7	10.7
Multi3	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Multi4	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Multi5	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Multi6	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Multi7	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0

[a]

[a] NS:The package doesn't converge in estimation.

Table 9.11: *Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in Matlab, Shazam, R, Minitab, Eviews, and SPSS for Testing Multicollinearity*

Proc	Matlab	Shazam	R	Minitab	Eviews(Default)	Eviews(NR)	SPSS
Parameter Estimates							
Base	13.81(13.95)	5.54(5.67)	11.68(12.01)	10.55(11.05)	9.00(9.49)	11.68(12.19)	10.57(11.13)
Multi1	9.45(9.58)	9.07(9.17)	10.39(12.09)	9.04(9.60)	6.29(9.49)	10.40(12.22)	7.91(8.00)
Multi2	10.70(11.15)	10.58(10.75)	11.06(11.46)	9.20(9.34)	7.41(7.53)	11.03(11.54)	9.33(9.35)
Multi3	10.75(11.73)	6.07(6.08)	10.83(11.32)	9.66(11.09)	8.92(8.94)	10.81(11.39)	10.84(11.27)
Multi4	10.43(12.43)	5.08(6.23)	10.41(11.64)	8.76(9.58)	7.30(7.86)	10.42(12.29)	10.43(12.44)
Multi5	10.46(12.02)	8.65(8.81)	8.97(9.23)	9.43(10.14)	7.79(9.62)	9.52(9.79)	10.45(12.02)
Multi6	10.83(12.29)	6.51(6.69)	10.83(12.25)	9.80(10.73)	7.80(8.20)	10.83(12.30)	10.83(12.21)
Multi7	10.53(11.51)	7.05(7.91)	10.53(11.50)	10.32(11.46)	10.53(11.52)	10.53(11.50)	10.54(11.60)
Standard Error Estimates							
Base	14.60(15.41)	3.15(3.56)	6.57(6.98)	6.23(6.64)	6.44(6.78)	6.76(7.18)	11.44(12.12)
Multi1	9.97(10.18)	4.98(5.20)	6.75(6.95)	8.51(8.77)	6.62(7.54)	6.37(6.64)	8.44(8.63)
Multi2	10.66(12.41)	5.46(5.55)	6.06(6.13)	4.74(4.83)	4.79(4.88)	5.96(6.04)	9.49(9.57)
Multi3	10.66(11.78)	2.90(2.91)	6.01(6.01)	5.04(5.04)	4.58(4.58)	5.83(5.83)	11.11(11.51)
Multi4	10.69(12.29)	2.93(3.61)	5.91(6.55)	4.91(5.54)	5.23(5.95)	6.29(6.92)	10.69(12.33)
Multi5	10.70(11.86)	4.62(4.99)	4.74(5.06)	7.86(8.14)	5.61(6.15)	5.02(5.34)	10.71(12.16)
Multi6	10.56(11.48)	3.52(3.73)	6.83(7.01)	6.01(6.19)	5.30(5.42)	7.19(7.37)	10.05(10.34)
Multi7	10.85(11.68)	3.40(3.96)	6.92(7.41)	5.88(6.38)	6.38(6.87)	6.76(7.25)	10.90(11.51)
Log-likelihood Estimates							
Base	10.53	6.26	10.57	10.0	10.50	10.50	10.57
Multi1	10.87	9.72	10.87	9.73	10.86	10.86	10.87
Multi2	10.72	11.24	10.72	10.21	10.77	10.77	10.72
Multi3	11.12	6.61	11.12	9.86	11.11	11.11	11.12
Multi4	11.18	6.49	11.18	10.41	11.19	11.19	11.18
Multi5	11.14	9.87	11.14	10.23	11.18	11.18	11.14
Multi6	12.05	7.81	12.05	10.26	11.00	11.00	12.05
Multi7	11.32	7.60	11.32	10.33	11.38	11.38	11.32

Table 9.12: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in Shazam, R, Minitab, and SPSS for Testing Multicollinearity*

Proc	Shazam-3	Shazam-5	Shazam-7	Shazam-9	Shazam-11	R1	R2	Minitab1	Minitab2	SPSS0,-6,-8	SPSS-5,-8,-10
Parameter Estimates											
Base	6.4	11.0	11.0	11.0	11.0	10.9	11.0	8.8	8.8	11.0	11.0
Multi1	9.1	11.0	11.0	11.0	11.0	7.5	9.1	9.0	9.0	7.9	7.9
Multi2	10.6	10.6	11.0	11.0	11.0	9.0	10.5	9.2	9.2	9.3	9.3
Multi3	6.1	11.0	11.0	11.0	11.0	10.0	10.7	9.6	9.8	10.8	10.8
Multi4	5.1	11.0	11.0	11.0	11.0	8.7	10.4	8.7	8.6	10.4	10.4
Multi5	8.7	11.0	11.0	11.0	11.0	7.2	9.0	9.4	9.0	10.5	10.5
Multi6	6.5	11.0	11.0	11.0	11.0	10.8	6.9	9.8	9.8	10.8	10.8
Multi7	7.1	11.0	11.0	11.0	11.0	10.6	10.5	9.5	9.5	10.5	10.5
Standard Error Estimates											
Base	3.6	7.2	11.0	11.0	11.0	5.8	7.3	5.8	7.3	11.0	11.0
Multi1	5.0	9.7	9.7	11.0	11.0	4.1	4.9	8.2	5.0	8.4	8.4
Multi2	5.5	5.5	10.7	10.7	11.0	4.6	5.4	4.7	4.7	9.5	9.5
Multi3	2.9	5.8	11.0	11.0	11.0	4.9	5.5	4.9	5.5	11.0	11.0
Multi4	2.9	5.6	11.0	11.0	11.0	4.7	5.8	4.7	5.8	10.7	10.7
Multi5	4.6	9.0	9.0	11.0	11.0	3.9	4.7	7.5	4.7	10.7	10.7
Multi6	3.5	6.9	11.0	11.0	11.0	5.8	3.7	5.8	5.8	10.1	10.1
Multi7	3.4	6.8	11.0	11.0	11.0	5.6	6.4	5.6	6.4	10.9	10.9
Log-likelihood Estimates											
Base	7.2	11.0	11.0	11.0	11.0	10.6	10.6	10.0	10.0	10.6	10.6
Multi1	9.7	11.0	11.0	11.0	11.0*	10.9	10.9	9.7	9.7	10.9	10.9
Multi2	11.0	11.0	11.0	11.0	11.0	10.7	10.7	10.2	10.2	10.7	10.7
Multi3	6.6	11.0	11.0	11.0	11.0*	11.0	11.0	9.9	9.9	11.0	11.0
Multi4	6.5	11.0	11.0	11.0	11.0*	11.0	11.0	10.4	10.4	11.0	11.0
Multi5	9.9	11.0	11.0	11.0	11.0	11.0	11.0	10.2	10.2	11.0	11.0
Multi6	7.8	11.0	11.0	11.0	11.0	11.0	11.0	10.3	10.3	11.0	11.0
Multi7	7.6	11.0	11.0	11.0	11.0	11.0	11.0	10.3	10.3	11.0	11.0

[a]

[a] *Warning message: log-likelihood function is decreasing. Indices 0,-3,-5,-7,-9,-11, and -13 following the software denote that convergence levels are 1E-3, 1E-5, 1E-7,1E-9,1E-11, and 1E-13.

Table 9.13: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in R and MINITAB for Testing Multicollinearity*

Proc	R(def)	R(1)	R(2)	R(3)	Minitab(def)	Minitab(1)	Minitab(2)	Minitab(3)
Parameter Estimates								
Base	11.0	10.9	10.0	11.0	8.8	8.8	8.8	8.8
Multi1	10.4	7.5	7.9	9.1	9.0	9.0	9.0	8.9
Multi2	11.0	9.0	9.6	10.5	9.2	8.9	9.4	9.7
Multi3	10.8	10.0	10.7	10.2	9.7	9.6	9.8	9.7
Multi4	10.4	8.7	9.4	10.4	8.8	8.7	8.7	8.7
Multi5	9.0	7.2	8.6	9.0	9.4	9.4	8.6	9.0
Multi6	10.8	10.8	10.8	6.9	9.8	9.8	9.8	9.8
Multi7	10.5	10.6	10.5	10.5	9.5	9.5	9.5	9.5
Standard Error Estimates								
Base	6.6	5.8	5.4	7.3	6.2	6.2	6.2	6.2
Multi1	6.7	4.1	4.4	4.9	8.5	8.2	8.6	4.9
Multi2	6.1	4.7	4.9	5.4	4.7	4.6	4.9	5.4
Multi3	6.0	4.9	5.5	5.0	5.0	4.9	5.5	5.0
Multi4	5.9	4.7	5.0	5.8	4.9	4.7	5.0	5.8
Multi5	4.7	3.9	4.6	4.7	7.9	7.5	4.6	4.8
Multi6	6.8	5.8	7.1	3.7	6.0	5.8	7.1	7.2
Multi7	6.9	5.6	6.0	6.4	5.9	5.6	6.0	6.4
Log-likelihood Estimates								
Base	10.6	10.6	10.6	10.6	10.0	10.0	10.0	10.0
Multi1	10.9	10.9	10.9	10.9	9.7	9.7	9.7	9.7
Multi2	10.7	10.7	10.7	10.7	10.2	10.2	10.2	10.2
Multi3	11.0	11.0	11.0	11.0	9.9	9.9	9.9	9.9
Multi4	11.0	11.0	11.0	11.0	10.4	10.4	10.4	10.4
Multi5	11.0	11.0	11.0	11.0	10.2	10.2	10.2	10.2
Multi6	11.0	11.0	11.0	11.0	10.3	10.3	10.3	10.3
Multi7	11.0	11.0	11.0	11.0	10.3	10.3	10.3	10.3

[a]

[a] NS:The package doesn't converge in estimation.

Table 9.14: *Minimum LREs for Parameter Estimates, Standard Errors, and Log-likelihood Function Changing Default Algorithm in Eviews for Testing Multicollinearity*

Proc	Eviews-Default-3	Default-5	Default-7	Newton Raphson-3	Newton Raphson-5	Newton Raphson-7	BHHH-3	BHHH
Parameter Estimates								
Base	8.6	8.8	8.8	8.8	8.8	8.8	7.6	7.6
Multi1	6.3	9.2	10.4	10.4	10.4	10.4	5.7	7.8
Multi2	7.4	10.8	10.8	11.0	11.0	11.0	5.3	7.2
Multi3	8.9	10.8	10.8	10.8	10.8	10.8	4.0	6.0
Multi4	7.3	7.3	7.3	10.4	10.4	10.4	5.0	7.3
Multi5	7.8	10.5	10.5	9.5	9.5	9.5	5.3	6.9
Multi6	7.8	10.9	10.8*	10.8	10.8	10.8	4.7	6.7
Multi7	10.5	10.5	10.5*	10.5	10.5	10.5*	3.8	3.8
Standard Error Estimates								
Base	6.4	10.1	10.1	6.8	6.8	11.0	1.4	1.4
Multi1	6.6	8.9	11.0	6.4	11.0	11.0	1.4	1.4
Multi2	4.8	7.7	10.7	6.0	10.8	10.8	1.3	1.4
Multi3	4.6	8.7	8.7	5.8	5.8	11.0	1.1	1.1
Multi4	5.2	8.1	8.1	6.3	10.7	10.7	1.2	1.2
Multi5	5.6	8.3	10.7	5.0	9.8	9.8	1.2	1.2
Multi6	5.3	8.5	10.1	7.2	7.2	10.7	1.6	1.5
Multi7	6.4	3.4	10.9	6.8	6.8	10.8	0.2	0.2
Log-likelihood Estimates								
Base	10.6	10.6	10.6	10.6	10.6	10.6	10.6	10.6
Multi1	10.9	10.9	10.9	10.9	10.9	10.9	10.9	10.9
Multi2	10.8	10.8	10.8	10.8	10.8	10.8	10.6	10.8
Multi3	11.0	11.0	11.0	11.0	11.0	11.0	7.5	11.0
Multi4	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Multi5	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Multi6	11.0	11.0	11.0	11.0	11.0	11.0	10.3	11.0
Multi7	11.0	11.0	11.0	11.0	11.0	11.0	6.7	6.7

[a]

[a] *More than 10 iterations. Indices following software algorithm -3, -5, and -7: Convergence levels are 1E-3, 1E-5, and 1E-7.

This decreasing trend of LRE accompanying with the increasing degree of multicollinearity implies the higher the degree of multicollinearity, the less reliable the standard error LREs (Table 9.11).

In Eviews, the difference between coefficient LREs and standard error LREs are significant, although this is not formally tested. The LRE results are less sensitive to the change of algorithm from default to Newton Raphson in the high multicollinearity, 2 covariate linear predictor case (multi5). The default algorithm - quadratic climbing hill depends on the functional form, because changing from quadratic, logarithm to pure linear setting enhances the LREs (i.e. multi6 vs. multi7, multi4 vs. multi5). However, the Newton Raphson algorithm results do not depend on the functional form. The reliability increases with the increase in the degree of multicollinearity, but an exception is that the Newton Raphson algorithm provides less accurate LREs for multi5 than multi4, although multi5 has a higher level of multicollinearity. This increasing trend of LRE is not causally related to the increase on degree of multicollinearity increases, although the covariates coherently contribute to the accurate estimations in this case. The loglikelihood LREs are independent of different algorithms used at different starting values, but does depend on the convergence criterion. The loglikelihood LREs increase from Multi1 to 3 (Table 9.11).

In Stata, "Binreg-BHHH" has unreliable standard error LREs. When estimating multi7 LREs go as high as 2 but are about 1 for many datasets. The Outer Product of the Gradient (OPG) standard error is different from Observed Information Matrix (OIM) in ML or Expected Information Matrix (EIM) of Maximum Quasi-likelihood (MQL). The MLE output of Newton Raphson is different from that for the BFGS algorithm. BFGS tends to provide the same LREs for standard error as Newton Raphson does in multi6 and even better than what Newton Raphson does in multi4. Binreg QML provides the best coefficient LREs for the multi1,2,3, but it loses its advantage to Logit and GLM MLE in multi4,5,6,7. MLEs of Logit and GLM provide the maximum standard error LREs for datasets multi1-5, but are lower than Binreg-BFGS for multi7 and Binreg-QML for multi6.

LREs depend on commands, estimators, and algorithms for different commands. For all Procs the Log-likelihood LREs decline from Multi1 to 3 and from Multi4 to 7 (Table 9.8).

In Limdep, although it has a central optimization routine for all procs, the LREs do not depend on the commands, but depend on the algorithm used. BFGS provides greater coefficient LREs than BHHH, except for multi1; and provides greater standard error LREs for multi2,3,4,7. When the degree of multicollinearity increases from multi1 to multi3, BFGS provides increasing LREs and BHHH produces decreasing LREs. When multiple functional forms are incorporated in the model, this relationship does not always hold. For example, from multi4 to multi5 the degree of multicollinearity increases, but the LREs decrease in BFGS; and with increasing multicollinearity from multi5 to multi6, LREs increase for the algorithm BHHH (Table 9.8).

In SAS, Logistic and Surveylogistic provide the same coefficient and Loglikelihood LREs, but different standard error LREs. The two Procs have the same coefficient estimator-MLE, but different standard error estimators: Logistic uses EIM but Surveylogistic uses linear approximation of estimator. Coefficient, standard error, and Loglikelihood LREs for the "Proc Logistic" "Proc Surveylogistic" and "Proc Catmod" increase with the increasing degree of multicollinearity from Multi1-3, but this relationship does not hold for multiple functional forms and models with more observations (e.g. Multi4-6). In model Multi4-6, the coefficient and standard error LREs decrease with an increasing degree of multicollinearity when the "Proc Catmod" is used. In the "Proc Genmod", the coefficient maximum LREs increase in Multi4-6, but the difference among coefficients decreases and standard error LREs are zero for Multi1-6. Coefficient minimum LREs for the "Proc Qlim" increase, but standard error minimum LREs decrease with an increasing degree of multicollinearity. "Proc Glimmix" provides zero Loglikelihood LREs for all datasets and the maximum coefficient LREs decrease with an increasing degree of multicollinearity from Multi1-3 and Multi4-6. Minimum standard error LREs increase and maximum standard error LREs decrease with an increasing degree of multicollinearity from Multi4-6 (Table 9.5).

9.8 Summary of Tests Based on Datasets Mvar1-4

SPSS provides the highest LREs (10 to 11) only for the Mvar1 but lower LREs (less than 10) for Mvar2-4. In Mvar3, the standard error LREs are greater than coefficient LREs. For Mvar4, the LREs improve when changing the convergence criterion. From Mvar2 to Mvar4, the loglikelihood LREs decrease. In all datasets, the coefficient LREs and standard error LREs are closely aligned (Table 9.21).

Matlab's performance is very steady for all datasets and provides consistently high (more than 10) LREs for coefficients and standard errors. A consistent result (as in SPSS) is that the LRE of the natural logarithm form of covariate is the largest for all coefficient LREs of Mvar1 (Table 9.21).

In R, the coefficient LREs are less than 10 except Mvar3. R does not provide an equivalently high LRE for standard errors, and only achieves a LRE value of 6 or 7. The Loglikelihood LREs are more than 10 (Table 9.21).

In Shazam, the coefficient LREs are very closely aligned with each other in one dataset estimation except in Mvar2. The standard error LREs are only about half of the coefficient LREs. From Mvar3 to Mvar4, all LREs decrease. From Mvar1 to Mvar2, most LREs increase (Table 9.21).

In Minitab, the deviation in LREs shows up in the different parameters of the models (e.g. Mvar1). The deviation in LREs also shows between coefficient and standard error LREs (e.g. Mvar4). Coefficient LREs are double the standard error LRE, but in certain datasets (e.g. Mvar1-3) the difference is not exaggerated. From Mvar3 to Mvar4, the maximum coefficient LRE, standard error and loglikelihood LREs decrease. From Mvar1 to Mvar2, the minimum coefficient LRE, standard error and loglikelihood LREs increase, but maximum coefficient LREs decrease. The loglikelihood LREs are very high regardless of the dataset (Table 9.21).

In Eviews, the difference between coefficient LREs and standard error LREs are significant, although it is not the main purpose of testing and been formally tested. LRE

Table 9.15: *Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in SAS for Testing Multivariate and Cut-Off Datasets*

Proc	SAS-Logistic(Fisher)	Logistic(NR)	Catmod	Genmod	Surveylogistic	Glimmix	Qlim
			Parameter Estimates				
Mvar1	6.62(6.87)	6.62(6.87)	10.61(12.90)	1.25(1.94)	6.62(6.87)	1.25(1.94)	7.96(8.19)
Mvar2	3.31(5.18)	3.31(5.18)	6.90(8.77)	0.42(1.62)	3.31(5.18)	0.43(1.62)	7.09(7.99)
Mvar3	3.62(3.82)	3.62(3.82)	7.28(7.51)	1.21(2.63)	3.62(3.82)	1.21(2.63)	8.30(9.10)
Mvar4	4.18(5.18)	4.17(5.18)	7.70(8.68)	NS	4.18(5.18)	0.05(0.20)	7.74(7.96)
Cuto1	2.42(4.26)	2.42(4.26)	0(0.62)	0.32(0.40)	2.42(4.25)	0.32(0.40)	4.87(6.99)
Cuto2	5.46(6.83)	5.46(6.83)	10.77(11.55)	0(0)	5.46(6.83)	0(0)	5.28(6.77)
Cuto3	6.75(7.32)	6.75(7.32)	NP	0(0)	6.74(7.32)	0(0.07)	5.68(7.67)
Cuto4	7.59(8.60)	7.59(8.60)	NP	NP	7.59(8.60)	0(0)	7.82(8.85)
			Standard Error Estimates				
Mvar1	7.08(7.29)	7.08(7.29)	6.73(6.93)	0(0)	1.24(1.92)	1.26(2.26)	5.37(6.59)
Mvar2	4.01(4.82)	4.01(4.82)	3.89(4.70)	0(0)	1.04(2.17)	0.86(1.30)	4.50(7.22)
Mvar3	4.59(4.95)	4.59(4.95)	7.97(8.33)	0(0)	1.87(2.72)	1.37(2.53)	5.68(7.46)
Mvar4	3.90(4.65)	3.90(4.65)	3.73(4.48)	NS	0.16(1.54)	0(0)	3.89(4.76)
Cuto1	4.38(4.78)	4.38(4.78)	0.07(0.25)	0(0)	0.10(0.14)	0.16(0.16)	4.34(4.54)
Cuto2	6.22(6.84)	6.22(6.84)	6.64(7.25)	0(0)	0.45(0.92)	0(0)	4.91(5.69)
Cuto3	7.01(7.49)	7.01(7.49)	NP	0(0)	0(1.20)	0(0)	4.08(4.82)
Cuto4	7.55(7.66)	7.55(7.66)	NP	NP	0.19(0.88)	0(0)	4.58(4.94)
			Log-likelihood Estimates				
Mvar1	10.56	10.56	10.56	0.05	10.56	0	10.56
Mvar2	8.45	8.45	11.63	0	8.45	0	11.63
Mvar3	8.90	8.90	11.32	0.24	8.90	0	11.32
Mvar4	8.93	8.93	11.17	NS	8.93	0	11.17
Cuto1	9.41	9.41	2.22	0	9.41	0	11.60
Cuto2	11.04	11.04	11.04	0	11.04	0.12	11.04
Cuto3	10.80	10.80	NP	0	10.80	0	10.79
Cuto4	11.29	11.29	NP	NP	11.29	0	11.31

[a]

[a] NS:not converge, NP:can not solve.

Table 9.16: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in SAS for Testing Multivariate and Cut-Off Datasets*

Proc	SAS-Logistic(Fisher1)	Logistic(Fisher2)	Logistic(NR1)	Logistic(NR2)	GEN1	GEN2	QLIM1	QLIM2	FISHER1X	FISHER2X	NR1X	NR2X	CAT,E
Parameter Estimates													
Mvar1	6.3	7.2	6.3	7.2	NS	NS	10.2	8.0	10.6	10.6	10.6	10.6	10.6
Mvar2	6.9	4.0	6.9	4.0	NS	NS	7.0	7.1	10.7	10.7	10.7	10.7	10.7
Mvar3	7.0	3.4	7.0	3.4	1.2	1.2	6.2	8.2	10.5	10.5	10.5	10.5	7.3
Mvar4	4.0	4.3	4.0	4.3	NS	NS	6.7	7.0	10.9	10.9	10.9	10.9	10.9
cuto1	2.0	2.0	2.0	2.0	0.3	0.3	5.9	5.0	9.2	9.2	9.2	9.2	0.0
cuto2	5.9	5.9	5.9	5.9	0	0	5.7	5.3	10.8	10.8	10.8	10.8	10.8
cuto3	7.3	7.3	7.3	7.3	0	0	7.8	6.1	10.7	10.7	10.7	10.7	10.7
cuto4	6.8	6.8	6.8	6.8	NS	NS	5.8	7.4	10.8	10.8	10.8	10.8	NP
Standard Error Estimates													
Mvar1	6.7	7.7	6.7	7.7	NS	NS	8.0	8.2	10.5	10.5	10.5	10.5	10.5
Mvar2	7.6	4.7	7.6	4.7	NS	NS	7.1	7.1	10.0	10.0	10.0	10.0	7.6
Mvar3	8.0	4.4	8.0	4.4	0	0	8.3	8.5	11.0	11.0	11.0	11.0	8.0
Mvar4	3.7	4.0	3.7	4.0	NS	NS	6.0	6.8	10.4	10.4	10.4	10.4	7.4
cuto1	3.9	3.9	3.9	3.9	0	0	6.4	5.3	10.4	10.4	10.4	10.4	0.1
cuto2	6.6	6.7	6.6	6.7	0	0	6.7	6.4	10.6	10.6	10.6	10.6	10.6
cuto3	7.5	7.5	7.5	7.5	0	0	4.1	4.1	10.4	10.4	10.4	10.4	10.4
cuto4	6.7	6.7	6.7	6.7	NS	NS	4.6	4.6	10.6	10.6	10.6	10.6	NP
Log-likelihood Estimates													
Mvar1	10.6	10.6	10.6	10.6	NS	NS	10.6	10.6	10.6	10.6	10.6	10.6	10.6
Mvar2	11.0	9.9	11.0	9.9	NS	NS	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Mvar3	11.0	8.5	11.0	8.5	0.2	0.2	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Mvar4	8.6	9.1	8.6	9.1	NS	NS	11.0	11.0	11.0	11.0	11.0	11.0	11.0
cuto1	8.6	8.6	8.6	8.6	0.0	0.0	11.0	11.0	11.0	11.0	11.0	11.0	2.2
cuto2	11.0	11.0	11.0	11.0	0	0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
cuto3	10.8	10.8	10.8	10.8	0	0	10.8	10.8	10.8	10.8	10.8	10.8	10.8
cuto4	11.0	11.0	11.0	11.0	NS	NS	11.0	11.0	11.0	11.0	11.0	11.0	NP

[a]

[a] NS:The package doesn't converge in estimation. NP: Can't solve the problem.

Table 9.17: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in SAS for Testing Multivariate and Cutoff Datasets*

Proc	SAS-Logistic(def)	Logistic(1)	Logistic(2)	Logistic(3)	Genmod(def)	Genmod(1)	Genmod(2)	Genmod(3)	QLIM(def)	QLIM(1)	QLIM(2)	QLIM(3)
Parameter Estimates												
Mvar1	6.6	6.3	6.6	7.2	1.3	NS	NS	NS	8.0	9.8	6.5	8.0
Mvar2	3.3	6.6	3.3	4.0	0.4	NS	NS	NS	7.1	7.0	7.6	7.1
Mvar3	3.6	7.0	7.1	3.4	1.2	1.2	1.2	1.2	8.3	6.2	8.1	8.4
Mvar4	4.2	4.0	4.4	4.3	NS	NS	NS	NS	7.7	6.1	7.3	8.4
cuto1	2.4	2.0	2.0	2.0	0.3	0.3	0.3	0.3	4.9	3.7	6.0	3.7
cuto2	5.5	5.9	5.9	5.9	0	0	0	0	5.3	5.7	6.2	5.3
cuto3	6.7	7.3	7.3	7.3	0	0	0	0	5.7	7.8	6.6	6.1
cuto4	7.6	6.8	6.8	6.8	NS	NS	NS	NS	7.8	5.8	5.7	7.4
Standard Error Estimates												
Mvar1	7.1	6.7	7.0	7.7	0	NS	NS	NS	5.4	5.4	5.4	5.4
Mvar2	4.0	7.6	4.0	4.7	0	NS	NS	NS	4.5	4.5	4.5	4.5
Mvar3	4.6	8.0	8.1	4.4	0	0	0	0	8.3	6.2	8.1	8.4
Mvar4	3.9	3.7	4.1	4.0	NS	NS	NS	NS	3.9	3.	3.9	3.9
cuto1	4.4	3.9	4.0	3.9	0	0	0	0	4.3	4.3	4.3	4.4
cuto2	6.2	6.6	6.7	6.7	0	0	0	0	4.9	4.9	4.9	4.9
cuto3	7.0	7.5	7.5	7.5	0	0	0	0	4.1	4.1	4.1	4.1
cuto4	7.6	6.7	6.7	6.7	NS	NS	NS	NS	4.6	4.6	4.6	4.6
Log-likelihood Estimates												
Mvar1	10.6	10.6	10.6	10.6	0	NS	NS	NS	10.6	10.6	10.6	10.6
Mvar2	8.4	11.0	8.4	9.9	0	NS	NS	NS	11.0	11.0	11.0	11.0
Mvar3	8.9	11.0	11.0	8.5	0.2	0.2	0.2	0.2	11.0	11.0	11.0	11.0
Mvar4	8.9	8.6	9.3	9.1	NS	NS	NS	NS	11.0	11.0	11.0	11.0
Cuto1	9.4	8.6	8.6	8.6	0	0	0	0	11.0	11.0	11.0	11.0
Cuto2	11.0	11.0	11.0	11.0	0	0	0	0	11.0	11.0	11.0	11.0
Cuto3	10.8	10.8	10.8	10.8	0	0	0	0	10.8	10.8	10.8	10.8
Cuto4	11.0	11.0	11.0	11.0	NS	NS	NS	NS	11.0	11.0	11.0	11.0

[a]

[a] NS:The package doesn't converge in estimation.

Table 9.18: *Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in Limdep and Stata for Testing Multivariate and Cut-Off Datasets*

Proc	Limdep(BFGS)	Limdep(BHHH)	Stata-Logit	-GLM	-Binreg	-Binreg(BHHH)	Binreg(BFGS)
Parameter Estimates							
Mvar1	10.61(12.62)	10.60(12.27)	7.61(8.34)	7.62(8.32)	7.62(8.31)	4.21(5.06)	4.60(5.72)
Mvar2	7.14(9.02)	8.78(10.05)	7.09(8.65)	7.17(8.87)	7.10(8.66)	3.08(5.22)	4.54(6.34)
Mvar3	8.15(8.39)	9.09(9.44)	7.29(7.98)	7.29(7.98)	7.29(7.98)	3.94(4.79)	9.37(9.65)
Mvar4	8.05(9.02)	6.55(7.77)	5.26(6.00)	4.58(5.48)	4.55(5.43)	2.61(3.89)	4.45(5.73)
Cuto1	5.80(7.62)	0.16(0.00)*	3.49(5.62)	3.49(5.62)	3.49(5.62)	1.15(3.39)	2.56(4.65)
Cuto2	5.46(6.83)	5.60(6.54)	6.13(8.07)	6.12(8.07)	6.11(8.13)	1.84(3.57)	4.74(5.67)
Cuto3	9.07(9.47)	9.64(10.92)	5.54(6.42)	5.81(7.19)	5.81(7.19)	2.54(5.23)	4.07(5.69)
Cuto4	8.31(9.08)	8.32(9.64)	5.45(5.96)	5.67(6.25)	5.66(6.25)	2.90(4.35)	5.08(6.98)
Standard Error Estimates							
Mvar1	10.48(12.33)	10.48(12.31)	6.97(7.38)	7.48(7.74)	4.91(5.15)	1.14(2.75)	5.32(5.59)
Mvar2	7.84(8.65)	9.13(11.39)	7.64(8.77)	7.70(9.51)	4.70(5.52)	0.90(1.50)	5.36(5.86)
Mvar3	9.15(9.52)	10.18(10.67)	7.43(9.44)	7.44(9.47)	6.39(6.74)	1.98(2.88)	7.59(9.06)
Mvar4	7.78(8.52)	6.50(8.30)	3.98(4.65)	3.69(6.02)	3.65(5.55)	0(0)	3.61(4.47)
Cuto1	7.75(8.16)	0.30(0.59)	6.04(7.02)	6.02(6.92)	4.99(5.32)	0(0)	5.03(5.80)
Cuto2	6.22(6.84)	6.38(7.20)	6.11(6.73)	6.74(8.51)	4.24(4.85)	0(0.17)	4.81(5.83)
Cuto3	9.16(9.63)	10.33(11.33)	5.69(7.03)	5.82(7.19)	4.80(5.33)	0(0.61)	5.07(5.32)
Cuto4	8.28(8.40)	8.55(8.57)	4.14(4.22)	5.23(6.67)	4.36(4.55)	0(0)	5.16(6.93)
Log-likelihood Estimates							
Mvar1	10.56	10.56	7.52	7.52	N/A	7.52	7.52
Mvar2	11.63	11.63	8.16	8.16	N/A	7.55	8.16
Mvar3	11.32	11.32	7.41	7.41	N/A	7.41	7.41
Mvar4	11.17	11.17	6.65	6.12	N/A	5.64	6.12
Cuto1	11.60	0.24*	6.99	6.99	N/A	6.26	6.99
Cuto2	11.04	11.04	7.56	7.56	N/A	6.45	7.55
Cuto3	10.80	10.80	6.81	6.70	N/A	6.50	6.70
Cuto4	10.97	10.97	6.70	6.79	N/A	6.39	6.79

[a]

[a] *Warning of flat log-likelihood function surface, N/A: not applicable.

Table 9.19: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in Limdep and Stata for Testing Multivariate and Cut-Off Datasets*

Proc	Limdep-BF1	-BF2	-BH1	-BH2	-BF1-13	-BF2-13	-BH1-13	-BH2-13	Stata-Binreg(Bh1)	-Binreg(Bh2)	Binreg(Bf1)	Binreg(Bf2)	Binreg(def)-13	G
Parameter Estimates														
Mvar1	10.6	7.2	10.6	10.6	10.6	10.6	10.6	10.6	4.2	4.2	4.6	4.6	7.6	
Mvar2	7.0	8.6	8.7	9.6	10.7	10.7	10.7	10.7	3.1	3.1	4.5	4.5	7.1	
Mvar3	7.0	7.7	7.5	9.8	10.5	10.5	7.5*	10.5	3.9	3.9	6.0	6.0	7.3	
Mvar4	7.7	8.2	7.6	9.8	10.9	10.9	10.9	9.8*	2.6	2.6	4.5	4.5	4.6	
Cuto1	5.7	5.7	7.4	7.0	9.2	9.2	9.2	9.2	1.1	1.1	2.6	2.6	3.5	
Cuto2	5.9	5.9	6.9	6.4	10.8	10.8	10.9*	10.8	1.8	1.8	4.7	4.7	6.1	
Cuto3	7.3	7.3	0 *	0 *	10.7	10.7	0 *	0 *	2.5	2.5	4.1	4.1	5.8	
Cuto4	6.8	6.8	7.8	7.7	10.8	10.8	10.8 *	10.8 *	2.9	2.9	5.5	5.1	5.7	
Standard Error Estimates														
Mvar1	10.5	7.7	10.5	10.5	10.5	10.5	10.5	10.5	1.1	1.1	5.3	5.3	7.5	
Mvar2	7.6	9.3	9.3	10.0	10.0	10.0	10.0	10.0	0.9	0.9	5.4	5.4	7.7	
Mvar3	8.0	8.7	8.4	11.0	11.0	11.0	8.4	11.0	2.0	2.0	7.6	7.6	7.4	
Mvar4	7.4	8.0	7.8	9.9	10.4	10.4	10.4	9.9	0	0	3.6	3.6	3.7	
Cuto1	7.7	7.7	8.8	8.7	10.4	10.4	10.4	10.4	0	0	5.0	5.0	6	
Cuto2	6.6	6.7	8.1	6.8	10.6	10.6	10.6	10.6	0	0	4.8	4.8	6.7	
Cuto3	7.5	7.5	0	0	10.4	10.4	0 *	0 *	0	0	5.1	5.1	5.8	
Cuto4	6.7	6.7	7.8	7.7	10.6	10.6	10.6	10.5	0	0	5.2	5.2	5.2	
Log-likelihood Estimates														
Mvar1	10.6	10.6	10.6	10.6	10.6	10.6	10.6	10.6	7.5	7.5	7.5	7.5	7.5	
Mvar2	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	7.5	7.5	8.2	8.2	8.2	
Mvar3	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	7.4	7.4	7.4	7.4	7.4	
Mvar4	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	5.6	5.6	6.1	6.1	6.1	
Cuto1	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	6.3	6.3	7.0	7.0	7.0	
Cuto2	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	6.5	6.5	7.6	7.6	7.6	
Cuto3	10.8	10.8	0	0	10.8	10.8	0 *	0 *	6.5	6.5	6.7	6.7	6.7	
Cuto4	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	6.4	6.4	6.8	6.8	6.8	

[a]

[a] *warning message: log-likelihood is flat at current estimates. *:Line search does not improve function, estimates.

Table 9.20: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in Limdep for Testing Multivariate and Cutoff Datasets*

Proc	Limdep-BFGS(def)	-BFGS(1)	-BFGS(2)	-BFGS(3)	-BHHH(def)	-BHHH(1)	-BHHH(2)	-BHHH(3)
Parameter Estimates								
Mvar1	10.6	10.6	10.6	7.2	10.6	10.6	10.6	10.6
Mvar2	7.1	6.9	7.0	8.6	8.8	8.7	8.8	9.6
Mvar3	8.1	7.0	7.1	7.7	9.1	7.5	10.5	9.8
Mvar4	8.0	7.7	8.5	8.2	6.5	7.6	7.5	9.8
Cuto1	5.8	5.7	5.8	5.8	FLAT	7.4	9.1	7.0
Cuto2	5.5	5.9	5.9	5.9	5.6	6.9	8.3	6.4
Cuto3	9.1	7.3	7.3	7.3	9.6	FLAT	FLAT	FLAT
Cuto4	8.3	6.8	6.8	6.8	8.3	7.8	7.7	7.7
Standard Error Estimates								
Mvar1	10.5	10.5	10.5	7.7	10.6	10.6	10.6	10.6
Mvar2	7.8	7.6	7.7	9.3	9.1	9.3	9.3	10.0
Mvar3	9.2	8.0	8.1	8.7	10.2	8.4	11.0	11.0
Mvar4	7.8	7.4	8.2	8.0	6.5	7.8	7.5	9.9
Cuto1	7.7	7.7	7.7	7.7	FLAT	8.8	10.3	8.7
Cuto2	6.2	6.6	6.7	6.7	6.4	8.1	9.1	6.8
Cuto3	9.2	7.5	7.5	7.5	10.3	FLAT	FLAT	FLAT
Cuto4	8.3	6.7	6.7	6.7	8.6	7.8	7.7	7.7
Log-likelihood Estimates								
Mvar1	10.6	10.6	10.6	10.6	10.6	10.6	10.6	10.6
Mvar2	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Mvar3	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Mvar4	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Cuto1	11.0	11.0	11.0	11.0	FLAT	11.0	11.0	11.0
Cuto2	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Cuto3	10.8	10.8	10.8	10.8	10.8	FLAT	FLAT	FLAT
Cuto4	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0

[a]

[a] NS:The package doesn't converge in estimation. FLAT:The log likelihood is flat at the current estimates.

Table 9.21: *Minimum (Maximum) LREs for Parameter Estimates, Standard Errors and Loglikelihood Function using Default Settings in Matlab, Shazam, R, Minitab, Eviews, and SPSS for Testing Multivariate and Cut-Off Datasets*

Proc	Matlab	Shazam	R	Minitab	Eviews(Default)	Eviews(NR)	SPSS
Parameter Estimates							
Mvar1	10.61(12.58)	7.62(7.90)	9.18(9.62)	9.42(10.61)	10.160(11.65)	8.58(9.04)	10.61(12.60)
Mvar2	10.71(11.84)	8.21(10.37)	8.52(10.69)	8.88(9.96)	7.62(10.25)	8.95(11.31)	7.17(9.04)
Mvar3	10.45(11.81)	9.37(9.65)	10.44(11.83)	7.27(9.74)	9.54(11.48)	10.45(11.81)	8.44(8.69)
Mvar4	10.94(11.85)	4.80(5.96)	9.56(10.72)	8.04(9.00)	8.36(9.35)	8.81(9.82)	8.34(9.32)
Cuto1	9.18(11.57)	6.67(8.77)*	7.85(9.28)	8.67(11.06)	9.18(11.30)	9.18(11.87)	6.27(8.07)
Cuto2	10.77(11.55)	6.28(7.85)	7.62(8.98)	9.41(10.32)	9.54(10.79)	9.60(10.77)	10.78(11.54)
Cuto3	10.67(11.26)	5.23(5.71)	9.42(9.86)	8.95(10.86)	9.52(9.97)	9.58(10.02)	10.67(11.25)
Cuto4	10.78(11.30)	9.56(10.44)	8.62(10.13)	7.95(10.70)	8.69(9.96)	8.69(10.17)	10.78(11.29)
Standard Error Estimates							
Mvar1	10.48(12.32)	4.14(4.37)	4.93(5.16)	7.08(7.28)	7.15(7.50)	4.62(4.90)	10.48(12.33)
Mvar2	9.98(12.34)	4.62(5.55)	4.70(5.52)	7.90(8.61)	4.80(5.73)	4.92(5.74)	7.86(8.68)
Mvar3	11.17(13.52)	5.28(5.69)	6.39(6.75)	7.21(8.53)	7.23(7.58)	8.11(8.48)	9.46(9.82)
Mvar4	10.42(11.43)	2.26(3.13)	4.66(5.41)	3.90(4.65)	4.06(4.81)	4.29(5.03)	8.08(8.83)
Cuto1	10.42(13.24)	4.57(5.04)	5.00(5.41)	8.18(8.63)	10.35(10.62)	10.57(10.80)	8.22(8.64)
Cuto2	10.65(11.18)	3.64(4.35)	4.25(4.85)	6.22(6.84)	5.24(5.84)	5.25(5.86)	10.64(11.17)
Cuto3	10.37(11.72)	2.77(3.32)	4.85(5.33)	6.98(7.46)	4.90(5.38)	4.92(5.40)	10.37(11.79)
Cuto4	10.59(12.11)	4.92(5.07)	4.42(4.55)	4.09(4.21)	4.46(4.59)	4.46(4.58)	10.60(11.83)
Log-likelihood Estimates							
Mvar1	10.56	8.33	10.56	10.45	10.60	10.60	10.56
Mvar2	11.63	9.72	11.63	11.63	11.00	11.00	11.63
Mvar3	11.32	10.24	11.32	10.45	11.34	11.34	11.32
Mvar4	5.72	5.81	11.18	10.22	11.12	11.12	11.17
Cuto1	11.60	9.90	11.60	9.79	11.50	11.50	11.60
Cuto2	11.04	8.08	11.04	10.20	11.04	11.04	11.04
Cuto3	10.80	6.19	10.80	10.35	10.74	10.74	10.80
Cuto4	10.97	10.20	10.97	10.35	11.02	11.02	10.97

Table 9.22: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in Shazam, R, Minitab, and SPSS for Testing Multivariate and Cut-Off Datasets*

Proc	Shazam-3	Shazam-5	Shazam-7	Shazam-9	Shazam-11	R1	R2	Minitab1	Minitab2	SPSS0,-6,-8	SPSS-5,-8,-10
Parameter Estimates											
Mvar1	7.6	11.0	11.0	11.0	11.0	10.6	10.6	9.4	9.4	10.6	10.6
Mvar2	8.2	11.0	11.0	11.0	11.0	6.9	8.6	9.3	8.6	7.2	7.2
Mvar3	9.4	11.0	11.0	11.0	11.0	10.5	7.7	7.3	7.3	8.4	8.4
Mvar4	4.8	9.3	11.0	11.0	11.0	7.7	8.2	7.7	8.2	8.3	8.4
Cuto1	6.7	10.6	10.6	10.6	10.6	5.7	5.7	8.7	8.7	6.3	6.3
Cuto2	6.3	11.0	11.0	11.0	11.0	10.8	10.8	9.4	9.4	10.8	10.8
Cuto3	5.2	10.4	11.0	11.0	11.0	7.3	7.3	9.0	9.0	10.7	10.7
Cuto4	9.6	11.0	11.0	11.0	11.0	10.8	10.8	9.9	9.9	10.8	10.8
Standard Error Estimates											
Mvar1	4.1	8.1	8.1	11.0	11.0	6.7	7.7	6.7	7.7	10.5	10.5
Mvar2	4.6	9.0	9.0	11.0	11.0	3.9	4.7	7.6	4.7	7.9	7.9
Mvar3	5.3	10.5	10.5	11.0	11.0	8.0	4.4	7.2	7.2	9.5	9.5
Mvar4	2.3	4.5	8.9	11.0	11.0	3.7	4.0	3.7	4.0	8.1	8.1
Cuto1	4.6	8.9	8.9	11.0	11.0	3.9	3.9	7.7	7.7	8.2	8.2
Cuto2	3.6	7.2	7.2	11.0	11.0	6.6	6.7	6.6	6.7	10.6	10.6
Cuto3	2.8	5.4	10.5	10.5	11.0	3.9	3.8	7.5	7.5	10.4	10.4
Cuto4	4.9	9.5	9.5	11.0	11.0	6.7	6.7	6.7	6.7	10.6	10.6
Log-likelihood Estimates											
Mvar1	8.3	11.0	11.0	11.0	11.0	10.6	10.6	10.5	10.5	10.6	10.6
Mvar2	9.7	11.0	11.0	11.0*	11.0*	11.0	11.0	11.0	11.0	11.0	11.0
Mvar3	10.2	11.0	11.0	11.0	11.0	11.0	11.0	10.5	10.5	11.0	11.0
Mvar4	5.8	10.3	11.0	11.0*	11.0*	11.0	11.0	10.2	10.2	11.0	11.0
Cuto1	9.9*	11.0*	11.0*	11.0*	11.0*	11.0	11.0	9.8	9.8	11.0	11.0
Cuto2	8.1	11.0	11.0	11.0	11.0	11.0	11.0	10.2	10.2	11.0	11.0
Cuto3	6.2*	11.0*	11.0*	11.0*	11.0*	10.8	10.8	10.4	10.4	10.8	10.8
Cuto4	10.2*	11.0*	11.0*	11.0*	11.0*	11.0	11.0	10.4	10.4	11.0	11.0

[a]

[a] *warning message: log-likelihood function is decreasing. Indices following software algorithm 0,-3,-5,-7,-9,-11 and -13 indicate convergence levels are 1E-3, 1E-5, 1E-7,1E-9,1E-11, and 1E-13.

Table 9.23: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Starting Values in R and MINITAB for Testing Multivariate and Cutoff Datasets*

Proc	R(def)	R(1)	R(2)	R(3)	Minitab(def)	Minitab(1)	Minitab(2)	Minitab(3)
Parameter Estimates								
Mvar1	9.2	10.6	10.6	10.6	9.4	9.4	9.4	9.4
Mvar2	8.5	6.9	7.0	8.6	10.7	10.7	10.7	8.6
Mvar3	10.4	10.5	10.5	7.7	7.3	7.3	7.3	7.3
Mvar4	9.6	7.7	8.5	8.2	8.0	7.7	8.4	8.2
Cuto1	7.8	5.7	5.8	5.7	8.7	8.7	8.7	8.7
Cuto2	7.6	10.8	10.8	10.8	9.4	9.4	9.4	9.4
Cuto3	9.4	7.3	7.3	7.3	9.0	9.0	9.0	9.0
Cuto4	8.6	10.8	10.8	10.8	7.9	9.9	9.9	9.9
Standard Error Estimates								
Mvar1	4.9	6.7	7.0	7.7	7.1	6.7	7.0	7.7
Mvar2	4.7	3.9	4.0	4.8	7.8	7.6	7.7	4.7
Mvar3	6.4	8.0	8.1	4.4	7.2	7.2	7.2	7.2
Mvar4	4.7	3.7	4.1	4.0	3.9	3.7	4.1	4.0
Cuto1	5.0	3.9	4.0	3.9	8.2	7.7	7.7	7.7
Cuto2	4.2	6.6	6.7	6.7	6.2	6.6	6.7	6.7
Cuto3	4.8	3.8	3.8	3.8	7.0	7.5	7.5	7.5
Cuto4	4.4	6.7	6.7	6.7	4.1	6.7	6.7	6.7
Log-likelihood Estimates								
Mvar1	10.6	10.6	10.6	10.6	10.5	10.5	10.5	10.5
Mvar2	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
Mvar3	11.0	11.0	11.0	11.0	10.5	10.5	10.5	10.5
Mvar4	11.0	11.0	11.0	11.0	10.2	10.2	10.2	10.2
Cuto1	11.0	11.0	11.0	11.0	9.8	9.8	9.8	9.8
Cuto2	11.0	11.0	11.0	11.0	10.2	10.2	10.2	10.2
Cuto3	10.8	10.8	10.8	10.8	10.4	10.4	10.4	10.4
Cuto4	11.0	11.0	11.0	11.0	10.4	10.4	10.4	10.4

[a]

[a] NS:The package doesn't converge in estimation.

Table 9.24: *Minimum LREs for Parameter Estimates, Standard Errors and Loglikelihood Function Changing Default Settings in Eviews for Testing Multivariate and Cut-Off Datasets*

Proc	Eviews-Default-3	Default-5	Default-7	Newton Raphson-3	Newton Raphson-5	Newton Raphson-7
Parameter Estimates						
Mvar1	10.2	10.2	10.6	8.6	10.6	10.6
Mvar2	7.6	10.7	10.7	8.9	10.7	10.7
Mvar3	9.5	9.5	10.5	10.5	10.5	10.5
Mvar4	8.4	10.9	10.9	8.8	10.9	10.9*
Cuto1	9.2	9.2	9.2	9.2	9.2	9.2
Cuto2	9.5	10.8	10.8	9.6	9.6	9.6
Cuto3	9.5	10.7*	10.7*	9.6	10.7*	10.7*
Cuto4	8.7*	10.8*	10.8*	8.7	10.8*	10.8*
Standard Error Estimates						
Mvar1	7.2	7.2	10.3	4.6	9.1	9.1
Mvar2	4.8	8.7	10.0	4.9	9.7	9.7
Mvar3	7.2	7.2	10.7	8.1	8.1	11.0
Mvar4	4.1	8.1	8.1	4.3	8.5	8.5
Cuto1	10.3	10.3	10.3	10.6	10.6	10.6
Cuto2	5.2	10.4	10.4	5.2	10.4	10.4
Cuto3	4.9	9.6	9.6	4.9	9.7	9.7
Cuto4	4.5	8.6	8.6	4.5	8.6	8.6
Log-likelihood Estimates						
Mvar1	10.6	10.6	10.6	10.6	10.6	10.6
Mvar2	11.0	11.0	11.0	11.0	11.0	11.0
Mvar3	11.0	11.0	11.0	11.0	11.0	11.0
Mvar4	11.0	11.0	11.0	11.0	11.0	11.0
Cuto1	11.0	11.0	11.0	11.0	11.0	11.0
Cuto2	11.0	11.0	11.0	11.0	11.0	11.0
Cuto3	11.0	11.0	11.0	11.0	11.0	11.0
Cuto4	11.0	11.0	11.0	11.0	11.0	11.0

[a]

[a] *more than 10 iterations. Indices -3,-5,-7 following each software algorithm denote convergence levels are 1E-3, 1E-5, 1E-7.

results are less sensitive to the change of algorithm from default to Newton Raphson in Mvar4. Newton Raphson provides higher LREs than the default in datasets Mvar2,3, and 4. The reason for default setting's extraordinary performance in Mvar1 needs to be investigated. After investigation, Mvar1 is a good benchmark model, which can be estimated reliably across all reasonable software packages (i.e. except SAS Glimmix Genmod, which are alternative estimates). The loglikelihood LREs are very high independently of choice of algorithms, but does depend on the convergence criterion. The loglikelihood LREs decrease from Mvar3 to 4 (Table 9.21).

In Stata, the standard error LREs are not reliable for Binreg-BHHH, (i.e. they are less than 4 in some cases). The OPG standard error is different from OIM in ML or EIM of MQL. In the same MLE output, NR is different from that of the BFGS algorithm. BFGS never provides the same standard error LREs as NR does for the Mvar datasets. Binreg MQL never provides the largest minimum coefficient LREs for the Mvar dataset, as it does in the Multicollinearity datasets. MLE of GLM provides the largest minimum coefficient and standard error LREs for datasets Mvar1,2 and 3, but they lose their advantage to MLE of Logit for Mvar4. LREs depend on commands and estimators, and the same algorithm for MLE for different commands provides different LREs. In all procs, the loglikelihood LREs for BHHH in Mvar2, 4 are the smallest. Logit GLM Binreg(BFGS) provides the same loglikelihood LREs, except in Mvar4 (Table 9.18).

In Limdep, the LREs do not depend on the commands, but depend on the algorithm used. BFGS provides larger coefficient LREs; and larger standard error and loglikelihood LREs in Mvar1,4 than BHHH; however the loglikelihood LRE provided by BFGS is smaller for Mvar1. When the variance of covariates decreases from 15 in Mvar3 to 1 in Mvar4, BFGS provides decreasing standard error LREs and BHHH produces decreasing LREs for coefficient, standard error, and Loglikelihood function. When multiple functional forms are incorporated in the model, such as from Mvar1 to Mvar2, the coefficient and standard error LREs decrease, but loglikelihood LRE increases for the algorithms BFGS and BHHH (Table

9.18).

In SAS, “Proc Logistic” and “Proc Surveylogistic” provide the same coefficient and loglikelihood LREs, but different standard error LREs. Coefficient, standard error, and loglikelihood LREs for “Proc Logistic”, “Proc Surveylogistic”, and “Proc Catmod” decrease, when incorporating different functional forms (e.g. Mvar1-2). In models, Mvar1-2, coefficient, standard error, and loglikelihood LREs of “Proc Logistic”, “Proc Surveylogistic”, and “Proc Catmod” increase when the variance of covariates decreases from 15 to 1. In “Proc Genmod”, the loglikelihood LREs decrease for Mvar1-2 and fail to converge for Mvar4. Coefficient and minimum standard error LREs in the “Proc Qlim” decrease, but loglikelihood LREs increase with an increasing number of covariates (Mvar1-2). “Proc Glimmix” provides zero loglikelihood LREs for all datasets and the decreasing coefficient and standard error LREs for Mvar1-2 and Mvar3-4 (Table 9.15).

9.9 Summary of All Tests Based on Datasets Cuto1-4

SPSS provides high LREs (10 to 11) for the Cuto2-4, but lower LREs (less than 10) for Cuto1. In Cuto1, the standard error LREs are greater than the coefficient LREs. From Cuto1 to Cuto3, the loglikelihood LREs decrease. In all the datasets, the coefficient LREs and standard error LREs are closely aligned but have a difference of up to 2 in Cuto1 (Table 9.21).

Matlab’s performance is very steady for all the datasets and consistently providing high (more than 10) LREs for coefficients, standard errors, and loglikelihood function, except the intercept in Cuto1. The coefficient and standard error minimum LREs increase, but coefficient, standard error maximum LREs and loglikelihood LREs decrease from Cuto1 to Cuto2. The coefficient, standard error, and loglikelihood LREs increase from Cuto3 to Cuto4 (Table 9.21).

In R, the coefficient LREs are relatively high about 8 and standard error LREs are not very high (less than 5). R can not provide an equivalently high LRE for standard error, and only achieve LREs of 6 or 7 as a maximum; however the loglikelihood LREs are consistently high (above 10). The minimum LREs and loglikelihood LRE decrease from Cuto3-4 (Table 9.21).

In Shazam, for all the parameters in the models, the coefficient LREs are very closely aligned, except in Cuto1. The standard error LREs are only about half of the coefficient LREs. From Cuto1 to Cuto2, all LREs decrease. From Cuto3 to Cuto4, all LREs in the models increase (Table 9.21).

In Minitab, the discrepancy in LREs in parameters for Cuto1 is up to 2. For Cuto4, coefficient LREs are double standard error LREs, but for Cuto1 the difference between coefficient and standard error LREs is very slight. From Cuto2 to Cut4, the minimum coefficient LRE decrease, and loglikelihood LREs increase. The loglikelihood LREs are very high regardless of the dataset (Table 9.21).

In Eviews, the difference between coefficient LREs and standard error LREs are significant for Cuto2-4. LRE results are less sensitive to what algorithm is used (i.e. default or NR). NR provides higher LREs than the default in datasets Cuto1-3, but lower LREs for Cuto4. The loglikelihood LREs are very high independently algorithmic choice. Loglikelihood LREs decrease from Cuto1-3, but increase from Cuto3-4. The minimum coefficient LREs decrease from Cuto2-4 and minimum standard error LREs decrease from Cuto1-4 (Table 9.21).

In Stata, the standard error LREs are not reliable for Binreg-BHHH, because their magnitudes are mostly zero. As the output of MLEs, the LRE output of NR is different from that of BFGS algorithm, and even the same NR provides different coefficient LREs in Logit and GLM for estimating Cuto3-4. Binreg QML provides the highest minimum coefficient LREs for the Cuto1,3 datasets, as it does in the Multicollinearity datasets; and provides the second lowest standard error LREs, which is only better than what BHHH does. MLE of GLM provides the highest coefficient and standard error LREs for datasets Cuto3 and largest standard error LREs for Cuto2. LREs depend on commands and estimators, and the same algorithm of MLEs in different commands provides different LREs. Among all procs the loglikelihood LREs of BHHH are the smallest for all datasets. Logit GLM Binreg(BFGS) provides the same loglikelihood LREs except for Cuto3,4, where loglikelihood LREs of Logit are slightly different from the other two (Table 9.18).

In Limdep, the LREs do not depend on the commands, but depend on the algorithm used. BHHH provides greater coefficient LREs, greater standard error LREs, and greater loglikelihood LRE for Cuto3-4 than BFGS; however only greater standard error and loglikelihood LREs for Cuto2. For Cuto1-2, BFGS provides decreasing LREs. For Cuto3-4, BFGS provides decreasing coefficient and standard error LREs, but increasing loglikelihood LREs. BHHH produces decreasing coefficient and standard error, but increasing loglikelihood LREs for Cuto3-4. BHHH fails to provide suitable LREs for Cuto1 and starting with values other than the default for Cuto3 gives a warning of a “flat” log-likelihood estimate, which indicates the MLE fails to achieve an optimum within a limited parameter range (Table 9.18).

In SAS, “Proc Logistic” and “Proc Surveylogistic” provide the same coefficient, and loglikelihood LREs, but different standard error LREs. Coefficient and standard error LREs of the Logistic proc, and the“Proc Surveylogistic” increase for Cuto1-4. Coefficient, standard error, and loglikelihood LREs of the “Proc Catmod” increase in Cuto1-2. In “Proc Genmod”, the coefficient, standard error, and loglikelihood LREs are zero, except coefficient LREs for Cuto1. In “Proc Qlim”, coefficient LREs increase for Cuto2-4, standard error LREs increase from Cuto1-2 and from Cuto3-4, but log-likelihood LREs decrease Cuto1-4. Glimmix provides zero log-likelihood LREs in Cuto1,3, and 4, zero coefficient and standard error LREs for Cuto2-4 (Table 9.15).

Chapter 10

Discussion

SAS - Logistic(Fisher or NR) performs reliably (LREs are more than 4)on 11 out of 16 models (Tables 13 and 23). For 2 out of 7 multicollinearity models, the LREs are less than 4, but the models do perform better for the null or OLS starting values (Table 15). Furthermore, increasing the tolerance level of the relative parameter convergence criteria to 1E-15 increases the reliability of these two models (Table 14). For 2 out of 4 multivariate models, the LREs are less than 4, but the models do better for the null or the closest starting values (Table 25). Furthermore, increasing the tolerance level of the relative parameter convergence criteria to 1E-15 increases the reliability for these two models (Table 24). For 1 out of the 4 cutoff models (i.e. Cuto1), the LREs are less than 4, and the model does not perform better by changing starting values (Table 25). However, increasing the tolerance level of the relative parameter convergence criteria to 1E-15 increases the reliability of Cuto1 (Table 24). SAS - Logistic estimates very well the standard error for all models except one multivariate model (i.e. Mvar4). SAS - Logistic estimates very well log likelihood function for all the models.

SAS - Catmod indicates that it is able to estimate successfully 12 out of 16 models (Tables 13 and 23). For 1 out of the 7 multicollinearity models, the LRE is nearly zero, but the model does not perform better when changing the tolerance level of the log-likelihood function to 1E-15. The estimations of standard errors are very reliable, except for 2 models, only one of which can be improved by changing the tolerance level of the log-likelihood function to 1E-15. Only one Log-likelihood function estimation is not reliable, but it can be improved by changing the tolerance level. For all multivariate models, the LREs are greater than 4, which means it estimates reliably in this series of model. However the estimation of standard errors are not as reliable as coefficients, 2 out of 4 models have unreliable standard error estimates. For 1 out of 4 cutoff models, the LRE is nearly zero and 2 out of 4 models can not be solved. For 2 out of them, changing the tolerance level of the log-likelihood function to 1E-15 can solve only one of the unsolved models. The estimation of log likelihood function is similar to the standard error in these four models.

SAS - Genmod, Glimmix can not reliably estimate any one of the 18 models. The unreliable result in Genmod can not be improved by changing starting values, because they use different approximated weight and frequency in the Log-likelihood function.

SAS - Surveylogistic can estimate reliably coefficients in 5 out of 7 multicollinearity models, 2 out of 4 multivariate models, and 1 out of 4 cutoff models; but the standard error estimates are not reliable. Changing the relative gradient tolerance level to 1E-15 can improve the coefficient estimations, but does not improve standard error estimations.

SAS - Qlim reliably estimates all models 16 out of 16 models except the unreliable standard error estimations 2 out of the 8 multicollinearity models and 1 out of the 4 multivariate models (Tables 13 and 23). Neither of them can be improved by changing the starting values from the default.

SAS performs reliably in “Proc Qlim” but not very steadily in “Proc Logistic”, “Proc Catmod”, and “Proc Surveylogistic”. SAS performs very unreliably in “Proc Genmod” and “Proc Glimmix”. To achieve a more reliable estimation, changing to OLS or zero may improve accuracy. Decreasing tolerance level from default to lower levels provide more reliable coefficient estimation, but does not always provide the the same for standard errors in “Proc Surveylogistic”.

Limdep - Logit and Blogit (BFGS) estimate all models reliably and they estimate only cutoff1 unreliably due to a flat functional surface around the optimum. The unreliable results can be improved by either using different starting values or decreasing the tolerance level to 1E-13.

Although Limdep shows some weakness in estimating parameters and standard errors in model cutoff1 using BHHH algorithm, it performs steadily well on most other models. Stata performs reliably for all models except parameter estimator in cutoff1 and standard error in multivariate4. However, Stata Binreg has some estimation deficiency when using the BHHH algorithm, especially for the multivariate and cutoff datasets.

Stata - Logit, GLM, and Binreg (default) estimate all models reliably except estimations of coefficients in cutoff1 and standard errors in multivariate4. The unreliable results can be improved by changing the gradient tolerance level to 1E-13. Binreg (BHHH) reliably estimates coefficient in 6 (5 in multicollinearity and 1 in multivariate models) out of 16 models, but reliably estimates standard error only in one multicollinearity model. Changing the tolerance level does not improve either the coefficient or the standard error estimates. Binreg (BFGS) estimates coefficients in 2 multicollinearity models and cutoff1 model unreliably, but standard error only unreliably in the model multivariate4. The result is not improved by using different starting values. The log likelihood functions can be estimated reliably in all models.

Shazam estimates the coefficient in all models reliably and the standard error in 8 out of 16 models reliably. In 7 multicollinearity models and one base model, the standard errors of 5 models can not be reliably estimated. In 4 multivariate models, it estimates only 1 model standard errors unreliably. In 4 cutoff models, standard errors in 2 models are estimated unreliably. These unreliable standard error estimations can be improved by changing the convergence level to 1E-5.

Unlike other packages, Shazam does not estimate reliable standard errors when the parameters are reliably estimated.

The rest of the packages are capable of estimating all models and show high reliability. Matlab, R, Minitab, Eviews, and SPSS reliably estimate all models.

Chapter 11

Conclusion

The reliability of nine software packages most commonly used by applied economists was examined. The packages were SAS 9.2, Limdep 9.0, Matlab 7.0.4, Stata 10, Shazam 10, R 2.10.1, Minitab 15, SPSS 17, and Eviews 3.1. Logistic regression using maximum log-likelihood estimation with nonlinear optimization algorithm tests were performed using developed benchmark datasets. This study expands on the existing literature in this area by examination of Minitab 15 and SPSS 17. The findings indicate that Matlab, R, Eviews, Minitab, Limdep (BFGS), and SPSS provided consistently reliable results for both parameter and standard error estimates across the benchmark datasets. While some packages performed admirably, shortcomings did exist. For example, SAS performed better than past versions, maximum log-likelihood estimators do not always converge to the optimal solution and stop prematurely depending on starting values, by issuing a "flat" error message. This quirk can be dealt with by rerunning the maximum log-likelihood estimator, using a closer starting point, to see if the convergence criteria are actually satisfied. Although Stata-Binreg provides reliable parameter estimates, there is no way to obtain standard error estimates in Stata-Binreg as of yet. Limdep performs relatively well, but did not converge due to a weakness of the algorithm. The results show that solely trusting the default settings of statistical software packages may lead to non-optimal, biased or erroneous results, which may impact the quality of empirical results obtained by applied economists.

The findings undermine the need to cross-validate empirical research results. The robustness and reliability of maximum log-likelihood estimates depend on the algorithm used, convergence level setting, the choice of starting values, the nature of the problem, and the datasets used in the problem. Researchers need to evaluate estimation results carefully until they are proven to be reliable, especially when multicollinearity, cut-off, and nonlinearities have mixed influences on the results. Learning econometrics requires an understanding of limitations of each software package and how to improve reliability of estimation results. Benchmarking and documenting the reliability of software packages are important tasks for software developers, but as an applied economist, performing reliability tests on the software latest versions is important for understanding and assessing a comprehensive software's capabilities.

Bibliography

Addelman, S. (1962) *"Orthogonal Main-Effect Plans for Asymmetrical Factorial Experiments"* Technometrics Vol. 4, pp.21-46.

Albert, A. and Anderson, J. (1984) *"On the Existence of Maximum Likelihood Estimates in Logistic Regression Models"* Biometrika 71, pp. 1-10.

Allison, P. D. (2008) *"Convergence Failures in Logistic Regression"* Presentation in SAS Global Forum 2008, P360-2008.

Altman, M. and McDonald, M. P. (2001) *"Choosing Reliable Statistical Software"* PS: Political Science and Politics XXXIV, pp. 681-687.

Altman, M., Gill, J. and McDonald, M. P. (2004) *Numerical Issues in Statistical Computing for the Social Scientist* John Wiley Sons, Inc..

Altman, M.m Gill, J., and McDonald, M., (2007) *"Accuracy: Tools for accurate and reliable statistical"* Computating Journal of Statistical Software. 21 (1), pp.1-30.

Amemiya, T. (1998) *Advanced Econometrics* Harvard University Press.

Anscombe, F. J. (1967) *"Topics in the Investigation of Linear Relations Fitted by the Method of Least-Squares"* Journal of the Royal Statistical Society B 29 pp.1-52.

Bangalore, S. S., Wang J. And Allison B.D. (2009) *"How Accurate are the Extremely Small P-values used in Genomic Research: An Evaluation of Numerical Libraries"* Computational Statistics and Data Analysis 53 pp. 2446-2452.

Bates, D. M. and Watts, D. G. (1988). *Nonlinear Regression Analysis and Its Applications* New York: John Wiley Sons.

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity* Wiley, New York.

Bergtold, J., Spanos, A. and Onukwugha, E. (2009) *"Bernoulli Regression models: Revisiting the Specification of Statistical Models with Binary Dependent Variables"* Journal of Choice Modeling. Vol.3 Issue 2, July 2010.

Berndt, E. (1990) *The Practice of Econometrics* MA: Addison-Wesley.

Brown, B., Lovato, J., Russell, R., (1997) *"DCDFLIB-Library of C routines for cumulative distribution functions, inverses, and other parameters"* Version 1.1.

Brown, B.W. (1998) *"DCDFLIB v1.1 (Double Precision Cumulative Distribution Function LIBrary)"* Available at ftp://odin.mdacc.tmc.edu/pub/source.

Burr, I.W. (1942) *"Cumulative Frequency Functions"* Annals of Mathematical Statistics 13:215-32.

Calzolari, G. and Panattoni, L. (1988) *"Alternative Estimators of FIML Covariance Matrix: A Monte Carlo Study"* Econometrica 56:3, pp.701-14.

Cline, A. K., Moler, C. B., Stewart, G. W., and Wilkinson, J. H. (1977) *An Estimate for the Condition Number of a Matrix, Argonne National Laboratory Research Report* No. TM-310.

Cody, P. R., Smith K. J., (2006) *Applied Statistics and the SAS Programming Language* Fifth Edition. Pearson Education, In..

Davidson, R. and MacKinnon, J. (2003) *Econometric Theory and Methods* Oxford University Press, New York.

Donaldson, J. R., and Schnabel, R. B. (1987) *"Computational Experience with Confidence Regions and Confidence Intervals for Nonlinear Least Squares"* Technometrics, 29, 67-82.

Estrella, A. (1998) *"A New Measure of Fit for Equations with Dichotomous Dependent Variables"* Journal of Business and Economic Statistics, Vol.16, No.2 (Apr. 1998), pp. 198-205. *Eviews 3 Manual User's Guide 2002*

Fiorentini, G., Calzolari, G., and Panattoni, L. (1996) *"Analytic Derivatives and the Computation of GARCH Estimates"* Journal of Applied Econometrics, 11, pp.399-417.

Fletcher, R. (1987). *Practical Methods of Optimization* 2nd edition. John Wiley Sons.

Fraley, C. (1989). *Software Performance on Nonlinear Least-Squares Problems* Report No. pp.89-1244, Stanford University Department of Computer Science.

Gill, P.E., Murray, W., and Wright M. H.(1981) *Practical Optimization* New York: Academic Press.

Gourieroux, C. (2000) *Econometrics of Qualitative Dependent Variables* Cambridge.

Greene, W. (2002) *Econometric Analysis* 5th Edition, Englewood Cliffs, H.J.: Prentice Hall.

Greenland, S., Schwartzbaum, J.A., and Finkle, W. D. (2000) *"Problems from Small Samples and Sparse Data in Conditional Logistic Regression Analysis"* American Journal of Epidemiology, 151, pp.531-539.

Harrell, Jr. F. E. (2001) *Regression Modeling Strategies: with Applications to Linear Models, Logistic Regression, and Survival Analysis* Springer-Verlag, New York, Berlin, Heidelberg.

Hauck, W. W. and Donner, A. (1977) *"Wald's Test as Applied to Hypotheses in Logit Analysis"* Journal of the American Statistical Association, Dec. 1977, 72 (360), pp. 851-53.

Hida, Y. Li, X. S., and Bailey D. H. (2001) *"Algorithms for quad-double precision floating point arithmetic"* . In Proc. 15th IEEE Symposium on Computer Arithmetic, IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 155-162.

Higham, N. J. (1996) *Accuracy and Stability of Numerical Mathematics* Philadelphia: SIAM.

Keeling, K. B., and Pavur, R. J. (2007) *"A Comparative Study of the Reliability of Nine Statistical Software Packages"* Computational Statistics and Data Analysis 51 pp.3811-3831.

Knsel, L. (1995) *"On the Accuracy of Statistical Distributions in GAUSS"* Computational Statistics and Data Analysis 20, pp.699-702.

Knsel, L. (1998) *"On the Accuracy of Statistical Distributions in Microsoft Excel"* Computational Statistics and Data Analysis 26, pp. 375-377.

Knsel, L., (2003) *Computation of statistical distributions documentation of program ELV* Second edition.

Le Cessie, S. and van Houwelingen (1997) *"Ridge Estimators in Logistic Regression"* Applied Statistics, 41, pp. 191-201.

Lesage, J.P., and Simon, S. D. (1985) *"Numerical Accuracy of Statistical Algorithms for Microcomputers"* Computational Statistics and Data Analysis, 3 pp. 47-57.

Li, X. S. Demmel, J. W, Bailey, D. H., Henry, G., Hida, Y. Iskandar, J. Kahan, W. Kapur, A., Martin, M. C.m Tung, T. And Yoo, D. J.(2000) *Design, Implementation and Testing of Extended and Mixed Precision BLAS.* Technical Reprot CS-00-451, Department of Computer Science, University of Tennessee, Knoxville, TN, USA, Oct. 2000. 61 pp. LAPACK working note 149.

Longley, J. W. (1967) *"An Appraisal of Computer Programs for the Electronic Computer from the Point of View of the User"* Journal of the American Statistical Association, 62, pp.819-841.

Maddala G. S. (1983) *Limited-dependent and Qualitative Variables in Econometrics* Cambridge.

McCullagh, P. and Nelder J.A. (1983) *Generalized Linear Models-(Monographs on statistics and applied probability)* Cambridge.

McCullough, B. D. (1997) *"Benchmarking Numerical Accuracy: A Review of RATS v.4.2"* Journal of Applied Econometrics, 12:2, pp. 181-190.

McCullough, B. D. (1998) *"Assessing the Reliability of Statistical Software: Part I"* The American Statistician, Vol. 52, No. 4 (Nov., 1998), pp. 358-366.

McCullough, B. D. (1999a) *"Econometric Software Reliability: Eviews, Limdep, Shazam, and Tsp"* Journal of Applied Econometrics. 14. Pp.191-202.

McCullough, B. D. (1999b) *"Assessing the Reliability of Statistical Software: Part II"* The American Statistician, Vol. 53, No. 2 (May, 1999), pp. 149-159.

McCullough, B. D. (2000) *"The Accuracy of Mathematica 4 as a Statistical Package"* Comutational Statistics, 15, pp.279-299.

McCullough, B. D. (2004) *"Some Details of Nonlinear Estimation"* In Altman, M., Gill, J. and McDonald, M. P., eds., Numerical Issues in Statistical Computing for the Social Scientist. John Wiley Sons, Inc..

McCullough, B. D. and Renfro, C. G. (1999) *"Benchmarks and Software Standards: A case Study of GARCH procedures"* Journal of Economic and Social Measurement, 25 (2) pp.59-71.

McCullough, B. D. and Renfro, C. G. (2000) *"Some Numerical Aspects of Nonlinear Estimation"* Journal of Economic and Social Measurement 26, pp. 63-77.

McCullough, B. D. and Vinod, H.D. (1999) *"The Numerical Reliability of Econometric Software"* Journal of Economic Literature, Vol. 37, No. 2 (Jun., 1999), pp. 633-665.

McCullough, B. D. and Vinod, H.D. (2003). *"Verifying the Solution from a Nonlinear Solver: A Case Study"* The American Economic Review, Vol. 93, No. 3 (Jun., 2003). Pp. 873-892.

McCullough, B. D. and Wilson, B. (1999) *"On the Accuracy of Statistical Procedures in Microsoft Excel 97"* Computational Statistics and Data Analysis 31 (1) pp.27-37.

McCullough, B. D. and Wilson, B. (2002) *"On the Accuracy of Statistical Procedures in Microsoft Excel 2000 and Excel XP"* Computational Statistics and Data Analysis 40 (1) pp.713-721.

McCullough, B. D. and Wilson, B. (2005) *"On the Accuracy of Statistical Procedures in Microsoft Excel 2003."* Computational Statistics and Data Analysis 49 pp.1244-1252.

McKenzie, C. (1998) *"A Review of Microsoft 4.0"* Journal of Applied Econometrics, 13:1 pp.77-89.

Meeker, W. Q. and Escobar, L. A. (1995) *"Teaching about Approximate Confidence Regions Based on Maximum Likelihood Estimations"* American Statistician, Feb. 1995, 49 (1), pp.48-53.

Monahan, F. J., (2001) *Numerical methods of statistics* Cambridge university press.

Moshier, S.L. (1989) *Methods and Programs for Mathematical Functions* Upper Saddle River, NJ: Prentice Hall. //www.netlib.org/cephes.

Myers, R. H., Montgomery, D. C., and Vining G. G. (2002) *Generalized Linear Model: with applications in engineering and the sciences* Jone Wiley Sons.

NIST Statistical Reference Datasets (StRD) website http://www.nist.gov/itl/div898/strd.

Nocedal, J. and Wright S.J (2000) *Numerical Optimization* 2nd edition. Springer Series in Operation Research and Financial Engineering.

Odeh, O., Featherstone, A. M., and Bergtold, S.J. (2010) *"Reliability of Economic Software"* American Journal of Agricultural Economics, forthcoming.

Oster, A. R. (2002) *"An Examination of Statistical Software Packages for Categorical Data Analysis Using Exact Methods"* The American Statistician, Vol. 56. No. 3 (Aug. 2002), pp. 235-246.

Oster, A. R. (2003). *"An Examination of Statistical Software Packages for Categorical Data Analysis Using Exact Methods-Part II"* The American Statistician, Vol. 57. No. 3 (Aug. 2003), pp. 201-213.

Oster, A. R. and Hilbe J. M. (2008). *"An Examination of Statistical Software Packages for Parametric and Nonparametric Data Analysis Using Exact Methods."* The American Statistician, Vol. 62. No. 1, pp. 74-84.

Overton, M. L. (2001) *Numerical Computing with IEEE Floating Point Arithmetic* Philadelphia: SIAM Press.

Renfro, C. G. (2004) *"A Compendium of Existing Econometric Software Packages"* Journal of Economic and Social Measurement 29, pp. 359-409.

Rogers, J., Filliben, J., Gill, L., Guthrie, W., Lagergren, E., and Vangel, M. (1998) *StRD: Statistical Reference Datasets for Assessing the Numercial Accuracy of Statistical Software* NIST TN1396, National Institute of Standards and Technology

Sawitzki, G. (1994a) *"Testing Numerical Reliability of Data Analysis Systems"* Computational Statistics and Data Analysis, 18, pp.269-286.

Sawitzki, G. (1994b) *"Report on the Reliability of Data Analysis Systems"* Computational Statistics and Data Analysis, 18, pp.289-301.

Simon, S. D., and Lesage, J. P. (1988) *"Benchmarking Numerical Accuracy of Statistical Algorithms"* Computational Statistics and Data Analysis, 7, pp.197-209.

Spanos, A. and McGuirk, A. (2002) *"The Problem of Near-multicollinearity Revisited: Erratic vs. Systematic Volatility."* Journal of Econometrics, 108, pp.365-393.

Stocks, Houston. (2004) *"On the Advantage of Using Two or More Econometric Software Systems to Solve the Same Problem"* Journal of Economic and Social Measurement, 29, 2004, pp. 307-320.

Train, K. E. (2003) *Discrete Choice Methods with Simulation* Cambridge.

Venables, W. N. and Ripley, B. D. (1999) *Modern Applied Statistics with S-plus* 3rd edition. New York: Springer. P.225.

Veall, M. R. (1991) *"Shazam 6.2: A review"* Journal of Applied Econometrics, 6, PP. 317-320.

Vinod, H.D. (1989) *"A Review of Soritec 6.2"* The American Statistician, 43:4. Pp. 266-269.

Vinod, H.D. (2000) *"Review of GAUSS for Windows Including its Numerical Accuracy"* Journal of Applied Econometrics. 15. Pp.211-220.

Vinod, H.D. and Ullah, Aman (1981) *Recent Advances in Regression Methods* New York: MarcelDekker.

Wampler, R. H. (1970) *“A Report on the Accuracy of Some Widely-Used Least Squares Computer Programs”* Journal of the American Statistical Association, 65, pp.549-565.

Wampler, R. H. (1980) *“Test Procedures and Problems for Least Squares Algorithms”* Journal of Econometrics 12 pp. 3-22.

Waren, A. D, Hung, M. S. And Lasdon, L. S. (1987) *“The Status of Nonlinear Programming Software: An Update”* Operations Research, Vol. 35, No. 4 (Jul. -Aug., 1987), pp. 489-503.

Yalta, A. T. (2007) *“The Numerical Reliability of Gauss 8”* American Statistician, 61, 3, pp. 262-268.

Yalta, A. T., and Jenal, O. (2008) *“On the Importance of Verifying Forecasting Results”* Working paper No:08-04. TOBB University of Economics and Technology Department of Economics.

Zellner, A., and T. Lee. (1965) *“Joint Estimation of Relationships Involving Discrete Random Variables”* Econometrica 33:383-94.

Appendix A

List of Acronyms

AIC: Akaike Information Criterion

BFGS or BF (in tables): Broyden Fletcher Goldfarb Shanno

BHHH or BH (in tables): Berndt Hall Hall Hausman

CDF: Cumulative Density Function

Cuto: Cutoff Datasets

DCDFLIB: Double Precision Cumulative Distribution Function LIBrary

DEF: Default

DFP: Davidon Fletcher Powell

EIM: Expected Information Matrix

ELV: Elementary Distribution

FCP: Fiorentini, Calzolari and Panattoni

FS: Fisher Scoring

GLM: Generalized Linear Model

EIM: Expected Information Matrix

GEE: Generalized Estimating Equations

IRLS: Iteratively Reweighted Least Squares

IWLS: Iteratively Reweighted Least Squares

LRE: Logarithm Reliable Error

MC: Monte Carlo

MLE: Maximum Likelihood Estimator

MQL: Maximum Quasi-Likelihood

Multi: Multicollinearity Datasets

Mvar: Multivariate Datasets

N/A: Not Applicable

NIST: National Institute of Standards and Technology

NP: Not Solved

NR: Newton Raphson

NS: Not Converge

OIM: Observed Information Matrix

OLS: Ordinary Least Squares

OPG: Outer Product of the Gradient

QUANEW: Quasi-Newton

SE: Standard Error

VIF: Variance Inflation Factor