Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

______________________________ ______________

Rachel Rutkowski Date

The Development of the CAT Statistical Injury Severity Score Incorporating Comorbidities, Age, and TRISS

By

Rachel Rutkowski

Master of Science in Public Health

Biostatistics

Patrick Kilgo

Thesis Advisor

The Development of the CAT Statistical Injury Severity Score Incorporating Comorbidities, Age, and TRISS

By

Rachel Rutkowski

B.S., B.A.

Case Western Reserve University

2007

Thesis Committee Chair: Patrick Kilgo, MS

An abstract of

A thesis submitted to the Faculty of the

Rollins School of Public Health of Emory University

in partial fulfillment of the requirements for the degree of

Master of Science in Public Health

in Biostatistics

2012

Abstract

The Development of the CAT Statistical Injury Severity Score Incorporating Comorbidities, Age, and TRISS

By Rachel Rutkowski

Background: Trauma severity scoring is a tool that standardizes the risk of a certain outcome, such as death, following a trauma incident. A universal, quantitative representation of this risk is critical to clinical treatment evaluation and the advancement of trauma research. The triage of patients, the assessment of hospital and clinic care, the prediction of trauma patient outcome performance, and epidemiologic studies all rely on trauma severity scoring. Despite the many advancements and promising applications of new severity scores, there is still a need for acceptance of these valuable tools by clinicians. There is a strong push for the statistical models to better account for comorbid conditions. Often, age is used as a proxy to represent the effect of pre-existing conditions as it is believed that many conditions like heart disease worsen or develop later in life. However, such logic does not always apply to all conditions or patients.

Methods: A sample of 1,250,549 patients and 39 variables was taken from the National Trauma Data Bank Version 7.1 between the Emergency Department admission years 2002 to 2006. Logistic Regression, multiple imputation, and model discrimination techniques (NRI/IDI) were employed to investigate the relationship between comorbidities, TRISS, and age through model development. Mortality risk ratios from model building analysis were used to create two Comorbidity-Age-TRISS (CAT) scores.

Comparisons between the CAT models, the unadjusted comorbidity model, and binary comorbidity model with TRISS and age were conducted.

Results: All models show a significant increased risk in mortality for a patient with pre-existing conditions; however, the score models show the highest risk - unadjusted model: RR=1.018, 95% CI=(1.018, 1.019); binary model: RR=1.018, 95% CI= (1.018, 1.019), unaltered CAT model: RR=1.273, 95% CI= (1.264, 1.283), beneficial CAT model: RR=1.185, 95% CI= (1.177, 1.192).

Conclusions: Comorbidities play a major role in the development of mortality risk. However, the extent to which these comorbidities affect mortality is still unclear. More research needs to be conducted on the conditions producing beneficial risk ratios (RR<1), and a unified approach to dealing with them in the creation of the scores should be employed.

The Development of the CAT Statistical Injury Severity Score Incorporating Comorbidities, Age, and TRISS

By

Rachel Rutkowski

B.S., B.A.

Case Western Reserve University

2007

Thesis Committee Chair: Patrick Kilgo, MS

An abstract of

A thesis submitted to the Faculty of the

Rollins School of Public Health of Emory University

in partial fulfillment of the requirements for the degree of

Master of Science in Public Health

in Biostatistics

2012

Dedication

To the love of my life, who has always given me his unconditional support, the reason why my seed for SAS was and will always be 05262007.

Acknowledgments

I'd like to thank all the folks in the BIOS department, especially Tracy, Bob, and Mary for always keeping me laughing even when things were stressful. Thank you to Melissa for helping me with the submission process even from states away. Thank you to my fellow students for working as a team and surviving the program together. I'd like to give a special shout out to Eric Hill for making sure I was completing my thesis and for keeping me on task. Thank you for your moral support. Thank you to Cara Mai at the CDC for your vote of confidence in my future career and for introducing me to the world of public health birth defects. Thank you to Dianne Stankiewicz (forever Miss. Maguire to me) for being the first person to make me think about a career in math. Thank you to David Bruckman from CWRU for introducing me to biostatistics and SAS. I still use his SAS notes. Thank you to Elizabeth Banks for providing me with my most amazing summer in Cleveland through AmeriCorps and for helping me realize that I want a career helping others.

I have so much appreciation and respect for my committee members Lance, George, and Pat. Thank you for your ongoing patience and support. Thank you especially to Pat Kilgo, my advisor, for taking a chance on me. From day one at Emory you have been my mentor and I am so fortunate to have had the opportunity to learn from you. Your kindness and understanding will not be forgotten.

Most of all, I'd like to thank my family (including all my kitties) and my wonderful husband. I am truly blessed to have you in my life. Thank you Mom and Dad for believing in me and for providing me with the opportunities to follow my dreams. I

really cannot sum up what you've done for me. Adam, you are an incredible husband and individual. Thank you for all your countless hours staying up helping me with homework and technical problems. Thank you for trusting in me and my dreams enough to agree to live separately, despite how hard, so I could go to graduate school in Atlanta and work in Tallahassee. Thank you for all the weekend drives and plane flights to see each other. You are my best friend and I am so lucky to call you my husband!

I tried my best to convey how thankful I am for the people I've encountered along my path to receiving my Masters but I am afraid that words cannot truly express how much their guidance, support, mentoring, and friendship have shaped my life. Perhaps John F. Kennedy said it best when he said, "As we express our gratitude, we must never forget that the highest appreciation is not to utter words, but to live by them." I will strive to continue to live by their example – to make result tables clear, save work early and often, to never stop learning, be kind to others, and above all, follow your heart.

I Eternally Thank You!

Table of Contents

Chapter One: Introduction……………………………………………………………...1

Background…………………………………………………………….…...1

Problem Statement……………………………………………………….....2

Purpose Statement…………………………………………………………..3

Significance Statement……………………………………………………...5

Chapter Two: Review of Literature……………………..…………………….....…7

Review of Anatomic Scores……………………………………….….......7

Review of Physiologic Status Scores………..…………………………….…....12

Review of Comorbidity Adjustment Scores...……..…………..…………….....12

Review of Combined Scoring Systems……………..…………………….…....13

Chapter Three: Research Methodology……………………………………….…15

Description of the Procedures……..……..…………………………….…....15

Data Cleaning & Variable Manipulations…………………...……..15

Methods for Data Investigation: Descriptive Statistics………..……..34

Methods for Data Investigation: Regression Analysis…...........……..36

CAT Severity Score Creation and Testing…...…………………...….43

Rationale for Solution Choice………………………………...……….…....47

Limitations……………...……………………………………………….…....47

Chapter Four: Results……………………………………………………………..…50

Description of the Outcome Results……...……………………………….…....50

Results of Data Investigation: Descriptive Statistics………..………..50

Results of Data Investigation: Regression Analysis…..............…….85

CAT Severity Score Testing Results……..…...……………………......102

Unexpected Problems and Findings……………..……………………….…..105

Chapter Five: Conclusions, Implications, and Recommendations…………………108

Conclusions……………..……………………………………………….…..108

Implications……………..…………………………………………...……..…..108

Recommendations……………..……………………..……………………..…..109

References……..……………………………………………………………….....112

Appendix A: SAS Code.……..……………………………………………......…115

Analysis Part 1.……..……………………………………...……………......…115

Analysis Part 2.……..……………………………………...……………......…117

Analysis Part 3.……..……………………………………...……………...…...119

Analysis Part 4.……..……………………………………...………...……...…128

Analysis Part 5.……..……………………………………...………...…...…129

Analysis Part 6.……..……………………………………...……………..……131

Analysis Part 7.……..……...…………………………...……………..……221

Analysis Part 8.……..……………………………………...……………......…240

Appendix B: SAS Macro...……..…………………………………………...……249

List of Tables

Table 1: *National Trauma Data Bank Datasets and Descriptions*……………….....…16

Table 2: *National Trauma Data Bank Sources for Main Variables*…………...…...…17

Table 3: *Main Variable Definitions and Variable Category Restrictions*…........……...22

Table 4: *Pre-existing Comorbidity Factors and Related ICD-9-CM Codes*……..……23

Table 5: *Pre-existing Comorbidity Variables and Descriptions*…………......……..….29

Table 6: *Main Variable Types, Missing Counts, and Missing Percentages*…...….......58

Table 7: *Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Categories*…………………………………….….59

Table 8: *Numerical Variable Ranges, Means, and Standard Deviations by Comorbidity Status*……………………………………..………………………….…73

Table 9: *Categorical Outcome Variable Counts and Percentages based on Total, within Age Range, and across Variable Category*………………………………….....…74

Table 10: *Numerical Outcome Variable Means and Standard Deviations by Age Range*………………………………………………………………………….……76

Table 11: *Comorbidity Counts and Percentages based on Total and Total Comorbidity*……………………………………………………………………...……77

Table 12: *Comorbidity Variable Counts and Percentages based on Total, within Age Range, and across Variable Category*…………………………………………….………79

Table 13: *Mortality Estimation Odds Ratios, Confidence Intervals, NRI, and IDI from Incremental Model Inclusions of Comorbidity, TRISS, and Age*……………...…92

Table 14: *Estimated Adjusted Comorbidity Effect on Length of Hospital Stay*……….96

Table 15: *Estimated Adjusted Comorbidity Effect on the Number of Patient ICU Days*……………………………………………………………..…….…………98

Table 16: *Estimated Adjusted Comorbidity Effect on the Number of Patient Ventilation Days*…………………………………..……………………….……...100

<u>Table 17</u>: ***CAT Severity Scoring System Comparison by Beneficial Odds Ratio Inclusion Status***……………………………………………………………….……….104

<u>Table 18</u>: ***Comparison of Mortality Estimation Odds Ratios, Confidence Intervals, NRI, and IDI Using Unadjusted Model, General Comorbidity Binary Model, and CAT Models*** ………………………………………………………………….……...105

<u>Table 19</u>: ***Comparison of Estimated Adjusted Comorbidity Effects on Outcome Variables Using the General Comorbidity Binary Model and CAT Model***...……......105

<u>Table 20</u>: ***Original NTDB Comorbidity Counts and Percentages***……....……...……..106

List of Models

<u>Model 1</u>: ***Unadjusted Logistic Regression Mortality Model* ………………………..…38**

<u>Model 2</u>: ***Logistic Regression Mortality Model with TRISS* ………………………..…38**

<u>Model 3</u>: ***Logistic Regression Mortality Model with TRISS and Age* ……………...…39**

<u>Model 4</u>: ***Regression Length of Hospital Stay Model* …………………………....…42**

<u>Model 5</u>: ***Regression Number of ICU Days Model* …………………………..….….42**

<u>Model 6</u>: ***Regression Number of Ventilation Days Model* …………………...……...43**

<u>Model 7</u>: ***Logistic Regression Mortality Model with Unaltered CAT Scoring Variable*…..……………………………………………………………………….…45**

<u>Model 8</u>: ***Logistic Regression Mortality Model with Beneficial CAT Scoring Variable*…………………………………………………………………………45**

<u>Model 9</u>: ***Regression Length of Hospital Stay Model with Unaltered CAT Scoring Variable* ……………………………………………………………..…46**

<u>Model 10</u>: ***Regression Number of ICU Days Model with Unaltered CAT Scoring Variable*..………………………………………….………………46**

<u>Model 11</u>: ***Regression Number of Ventilation Days Model with Unaltered CAT Scoring Variable*……………………………...…………………………………......46**

<u>Model 12</u>: ***Regression Length of Hospital Stay Model with Beneficial CAT Scoring Variable*……………………………………………………………………...46**

<u>Model 13</u>: ***Regression Number of ICU Days Model with Beneficial CAT Scoring Variable*……………………………………………………………………...46**

<u>Model 14</u>: ***Regression Number of Ventilation Days Model with Beneficial CAT Scoring Variable*…..………………………………………………………...….46**

List of Examples

Example 1: ***Data Format after the Initial Merging of NTDB Datasets rdsed and rdscomorbid*……………………………………….……………………………..…26**

Example 2: ***Data Format after the Reduction to only include Facilities and ED Years that Provide Comorbidities*…………………………………………….……………..…26**

Example 3: ***Step 1 of the Creation of Horizontal, Single Record, Binary Comorbidity Variables.*……………………………………………………………….……………..…28**

Example 4: ***Step 2 of the Creation of Horizontal, Single Record, Binary Comorbidity Variables.*…………………..……………………………………………..…………..…28**

Example 5: ***Step 3 of the Creation of Horizontal, Single Record, Binary Comorbidity Variables.*………………………………………………………………………………..…30**

Example 6: ***Step 4 of the Creation of Horizontal, Single Record, Binary Comorbidity Variables.*……………………………………………..…………………..……..…31**

Example 7: ***Step 5 of the Creation of Horizontal, Single Record, Binary Comorbidity Variables.*…………………………………………..…………………..……..…32**

Chapter One: Introduction

Trauma severity scoring is a tool that quantifies and standardizes the risk of a certain outcome, such as death, following a trauma incident. A universal, quantitative representation of this risk is critical to clinical treatment evaluation and the advancement of trauma research. The triage of patients, the assessment of hospital and clinic care, the prediction of trauma patient outcome performance, and epidemiologic studies all rely on trauma severity scoring. Its application has broadened to the business realm as well. "In the future, such scoring systems will play a larger role in financial reimbursement or even accreditation for individual critical-care units" (Rennie and Brady 2007).

BACKGROUND

Although the study of trauma is not a new field, the development and implementation of trauma severity scoring is relatively young. Improvements in technology and the development of new statistical techniques have made it possible to analyze large trauma datasets which were previously too cumbersome to handle adequately. The adoption of injury coding and the training of personnel for transcribing clinical records relating to trauma have been invaluable to the creation of these scoring systems. The Abbreviated Injury Scale (AIS) and the International Classification of

Diseases (ICD) are examples of injury coding frequently used. AIS, especially, has formed the basic component of many injury scoring systems.

Trauma severity scores can be used for risk adjustment. “In health services research, risk adjustment entails the identification of inherent factors that affect outcome independent of medical care” (Clark and Winchell 2004). Common “inherent factors” include age, gender, and pre-injury medical conditions, or comorbidities. Isolating such intrinsic patient factors allows for the examination of other potential risk sources such as the quality of hospital care. Trauma severity scores can be grouped into three distinct categories of risk adjustment: Anatomic Injury Scores, Physiologic Status Scores, and Comorbidity Scores (Kilgo, Meredith et al. 2008). It is also possible to have a severity scoring system that accounts for several of these categories. One of the most notable examples of this type of scoring system is the Trauma and Injury Severity Score (TRISS), which will be discussed in detail in future chapters.

PROBLEM STATEMENT

Despite the many advancements and promising applications of severity scores, there is still a need for acceptance of these valuable tools by clinicians. Many studies have shown that there is a high specificity for predicting the mortality of Intensive Care Unit (ICU) patients using these scoring systems, but clinically, perhaps a high specificity is not enough. Only approximately 55% of the variance in survival outcomes is accounted for by noteworthy scoring systems (Kilgo, Meredith et al. 2008). “It is an unfortunate fact that given the importance of the decisions to be made based on mortality prediction, clinicians would be unlikely to adopt such a system unless it is virtually perfect”

(Gartman, Casserly et al. 2009). For this reason, severity scoring is rarely used in acute settings for clinical decision-making. There is clearly a need for model methodology improvements.

There is a strong push for the statistical models to better account for comorbidities, which are a patient's preexisting medical conditions. "Comorbidity scoring is essential for calculating risk adjusted outcomes in trauma similar to the risk adjustment outcomes that exist for cancer and heart disease" (Fields 2008). Often age is used to represent the effect of pre-existing conditions as it is believed that many conditions like heart disease worsen or develop later in life. However such logic does not always apply to all conditions or patients. Comorbidities like chronic drug abuse or previous spinal cord injuries can occur at younger ages. "Preexisting condition status has consistently been found to be an independent predictor of mortality after trauma. However, a preexisting condition index has not yet been validated for use in a trauma population" (Moore, Lavoie et al. 2008).

In addition to improving methodology by strengthening comorbidity scoring, outcomes other than mortality require further investigation. So much effort has been focused on mortality that quality of life factors are still in question. Injury severity scoring could be incredibly useful in predicting outcomes like a patient's length of stay in the hospital or his or her Functional Independence Measure (FIM) at discharge. However, there are often large quantities of missing information in trauma datasets that hinder the strength of these scoring systems. Great care should be applied in statistically analyzing such incomplete records so as not to lose pertinent data.

PURPOSE STATEMENT

I have devised a trauma severity scoring system that aims to address the current gap in knowledge as stated in the problem statement. The research data was acquired from the National Trauma Data Bank (NTDB) and contains extensive patient data from over 900 United States trauma centers spanning from 2002 to 2007. Multivariable linear and logistic regression models were constructed to evaluate the effect of trauma on four outcomes: mortality, length of patient hospital stay, the number of days in the ICU, and the number of days a patient is on ventilator support. Forty-five comorbidities, as well as the overall presence of any comorbidities, were recorded as either absent or present. The project objectives were as follows:

(1) Estimate the additional risk of mortality after trauma using the incremental inclusion of comorbidities, TRISS, and age in logistic regression models.

(2) Estimate the adjusted comorbidity effect on the length of hospital stay, the number of ICU days, and the number of days a patient requires ventilator support using regression analysis.

(3) Created the Comorbidity-Age-TRISS (CAT) trauma severity scoring system based on the estimated risk of mortality from the full multivariable logistic regression model in objective (1).

This analysis will have three methodological advantages over current metrics. First, the use of TRISS placed in the risk adjustment model for mortality alongside comorbidity and age, captures physiological and anatomical criteria. Adding all three predictors into the model creates a more encompassing combined scoring system. Second, the study was not hampered by missing information. It was accounted for by the statistical procedure multiple imputation (MI) that allows for the inclusion of patient data in scores even if it is incomplete. And lastly, two newer methods of assessing improvement in logistic model performance were employed - the Net Reclassification Improvement (NRI) and the Integrated Discrimination Improvement (IDI). These methodologies are advantageous over the traditional p-values indicators given the large size of the NTDB dataset. The use of TRISS, MI, and NRI/IDI will be discussed in more detail in later chapters.

SIGNIFICANCE STATEMENT

"Almost two million Americans are hospitalized each year due to trauma, and 45,000 (2.4%) of these patients die before discharge" (Osler, Glance et al. 2008). Although trauma centers and hospitals have no doubt saved many lives, there is likely a large variation in outcomes across these institutions. The ability to accurately and systematically evaluate these trauma centers is critical to improving the health care that is dispersed. The creation of a risk adjustment model that builds on current trusted scores and better incorporates comorbidities can only aid in achieving more effective health care programs. Indeed, the end goal is to save as many patients as possible.

It is not just important to improve health care programs to rectify procedures for those that could have lived but to also identify those that are critically ill so that the patient's end-of-life care is managed as correctly and humanely as possible. "Many studies have consistently shown that healthcare expenditures on dying patients are disproportionately high and recent data show that use of high cost and invasive therapies on patients who die is increasing" (Gartman, Casserly et al. 2009). One of the most pivotal components for quality end-of-life care is for the hospital staff to know as early as possible that the patient is highly unlikely to benefit from resource intensive and possibly painful procedures. Trauma severity scores could be one tool used to assist in this decision.

Chapter Two: Review of Literature

There are numerous sources of information about trauma and trauma severity scoring that have been published or made available. Although hundreds of these sources are relevant to this project, only a handful was chosen to be reviewed. Literature was selected on the basis of the source reputation, contribution to the development of other scoring systems, the notoriety of the scoring system addressed, and its focus on a clear category of risk adjustment. The following review of literature is organized by subheadings for the category of risk adjustment as anatomic injury scores, physiologic status scores, comorbidity scores, or combined scoring systems. Within each category, journals were reviewed chronologically.

REVIEW OF ANATOMIC INJURY SCORES

The first anatomic injury coding lexicon was published in 1971 by the American Medical Association Committee on Medical Aspects of Automotive Safety. The score was termed the Abbreviated Injury Scale (AIS) and is a numerical scale ranging from 1 to 6 representing the degree of injury severity. The classifications were determined by clinical experience and were labeled as 1 for “minor”; 2 for “moderate”; 3 for “severe, not life-threatening”; 4 for “severe, life-threatening, survival probable”; 5 for “critical, survival uncertain"; and 6 for "fatal" (NTDB 2008). This numerical score was determined based on the injuries to each of the following five body regions - head or neck, chest, abdomen, pelvis/extremities, and general region. There were later several revisions to AIS which expanded the number of body region categories from five to nine and created specific coding for pediatric injuries. The initial purpose of AIS was to describe isolated blunt trauma but when applied to assessing multiple traumas it was found to be less useful (Linn 1995).

To address patients with multiple traumas, Baker et al. developed a method based on AIS that they called the Injury Severity Score or ISS. Their findings were first presented in 1974 in the Journal of Trauma under the title "The Injury Severity Score: A Method for Describing Patients with Multiple Injuries and Evaluating Emergency Care". ISS was adapted from the original AIS by adding face as a sixth body region of injury area classification and readjusting the severity range of 1 to 6 where 1 represents a "minor injury" and 6 represents an "unsurvivable injury". ISS is computed by taking the sum of squares of the AIS from the three most severely injured body regions. If any body region sustained an AIS rated injury of 6, or the maximum injury level, then the ISS automatically becomes a score of 75. Therefore ISS is only able to range from 1 to 75.

"Injury Severity Score represents an important step in solving the problem of summarizing injury severity, especially in patients with multiple trauma" (Baker 1974). ISS is clearly the most applied anatomic injury scoring system and it correlates strongly with mortality; however, it is not without limitations. "ISS is most often treated as a continuous, ordinal, monotonic function of mortality but it is in fact none of these" (Kilgo 2008). It is possible for two separate combinations of body injury region AIS sum of squares to produce to same ISS score. Two independent AIS combinations should ideally result in two separate scores. Also ISS does not distinguish between multiple injuries sustained to the same body region. Future anatomic injury scores were developed with the aim of rectifying these ISS limitations.

Copes et al. in *The Journal of Trauma* publication "Progress in Characterizing Anatomic Injury" in 1990 proposed the concept of a four-valued descriptor of patient trauma called Anatomic Profile (AP). AP aims to address the limitations of ISS by accounting for multiple injuries to the same body region and better accounting for severity level differences. The body regions were defined as "head/brain and spinal cord", "Thoracic and Front of Neck", and "All Others". The top three AIS scores were taken for each body region given that the injuries were deemed as "severe" or AIS greater than 2. Composite scores of the three AIS scores per body region were then labeled as A, B, and C respectively. A, B, and C, along with the overall maximum AIS score (maxAIS) for all regions became the four descriptors creating the AP scoring system. Although AP performed with a 31% increase in sensitivity over ISS in model performance comparisons, like ISS it cannot reliably predict patient outcome. AP has yet to gain popularity in use.

Osler et al., like the inventors of the AP score, also attempted to remedy ISS's multiple injury per body region issue in the 1996 publication "ICISS: An International Classification of Disease-9 Based Injury Severity Score". They additionally were wary of ISS's dependence on AIS lexicon stating that such a coding system is expensive, excluding all but the hospitals with a "zealous commitment to trauma care". Considering these limitations, Osler et al. devised the ICD-9 Injury Severity Score (ICISS) which is based on the more economical International Classification of Disease 9th Edition (ICD-9). ICISS is the product of the Survival Risk Ratios (SRR) of each of a patient's injuries. SRRs are derived for all ICD-9 800 through 959.9 injury categories by dividing the number of patients that survive a given injury by the number of patients recorded as displaying that given injury. Despite the allowance of an AIS free scoring system, ICISS has several intrinsic issues using database driven information. The score is only as accurate as the database it renders from.

Following Osler's publication describing ICISS, in 1997 Osler et al. published "A Modification of the Injury Severity Score That Both Improves Accuracy and Simplifies Scoring". In it they describe a variation of the ISS called the New Injury Severity Score (NISS). NISS aims to eliminate the imposed one injury to each body region limitation ISS has by taking the sum of the squares of a patient's three most severe AIS scores independent of body region. This adaptation showed an improvement in prediction but was unable to eradicate several of ISS's original limitations.

With several scoring systems to choose from, evaluation needed to be performed to assess the score that best predicts patient survival past each individual score evaluation. Dr. William Sacco and his colleagues were the first to attempt to compare AIS, ISS, AP, ICISS, and NISS using a common database and standardized measure of score performance. In 1999 they published the results of this comparison in *The Journal of Trauma* as "Comparison of Alternative Methods for Assessing Injury Severity Based on Anatomic Descriptors". Using data from the Pennsylvania Trauma Outcome Study they found that AP, a modified version of AP, and NISS performed better in predicting patient survival than the ICD-9 based scoring system, ICISS. Their findings also indicated that the modified AP, AP, and NISS are preferable as a measure of overall severity to ISS.

Meredith et al. in 2002 also performed a joint comparison of nine AIS and ICD-9 based scoring systems to evaluate their ability in predicting mortality. AIS-based ISS, NISS, AP, and maxAIS, as well as these four scores when mapped from ICD-9, were considered along with the ICD-9 based ICISS. Using data from the National Trauma Data bank (NTDB) they found minor differences in performance between scores. However, ICISS and AP showed the best improvement in discrimination compared with ISS and maxAIS, and AIS based scores performed better than ICD-9 mapped based scores.

Kilgo et al. in 2003 took the comparison process a step further by including the possibility of only a patient's worst injury in predicting outcome. Using the NTDB, six scores were computed and compared for survival discrimination, model fit, and explainable variance proportion. The scores were ICISS, ICISS1 – patient's worst ICD-9 survival risk ratio, Trauma Registry Abbreviated Injury Scale Score (TRAIS), TRAIS1 – patient's worst AIS survival risk ratio, NISS, and maxAIS. It was found that the scores that used only a patient's worst injury, maxAIS, TRAIS1, and ICISS1, out-performed their counterparts.

In 2008, Osler and colleagues also attested that ISS underperforms compared to a patient's single worst injury. Osler devised three new models created from empirical estimates of individual injury severity to compare to ISS. One of his models in particular, the Trauma Mortality Prediction Model (TMPM) had better discrimination and calibration than ISS especially when the models were augmented with age, gender, and mechanism of injury.

REVIEW OF PHYSIOLOGIC STATUS SCORES

The first medically accepted physiologic status score was the Glasgow Coma Scale (GCS) created by Teasdale et al. for tracking physiologic functionality in patients. The scale is composed of three components for eye, motor, and verbal ability. Each component is assigned a ranking from one to four for eye, one to five for verbal response, and one to six for motor response representing the level of functionality. All three components can be aggregated into the total GCS score and ranges from unresponsive to responsive (3 to 15). GCS was quickly accepted and continues to be the most widely used scoring system.

Champion et al. created another physiologic score called the Trauma Score (TS) in 1983 and later revised it in 1989 renaming it to the Revised Trauma Score (RTS). TS was revised to include GCS, respiratory rate, and systolic blood pressure. It is a weighted sum of these variable indices producing a score that ranges from 0 to 7.84 where 7.84 is the highest functionality score. Unlike GCS however, RTS is seldom employed in practice.

REVIEW OF COMORBIDITY SCORES

In 1987, Dr. Charlson and colleagues proposed a new method for classifying comorbidities for mortality risk assessment – the Charlson Comorbidity Index (CCI). The CCI is a collection of 19 pre-existing conditions which are weighted by their mortality association. The score was developed from a cohort of 604 patients admitted into New York Hospital in the same month in 1983 and examined for one-year. The score was then validated on a cohort of 685 patients admitted to Yale New Haven Hospital in 1962 to 1969. CCI was not however validated on trauma populations.

Morris et al. in 1990 conducted a case-control study with 1983 California acute care hospital data to examine the effects of comorbidity on hospital mortality. Eleven pre-existing conditions were studied and five were found to significantly increase the risk of mortality. They were cirrhosis (OR = 4.5), congenital coagulopathy (OR = 3.2), ischemic heart disease (OR = 1.8), chronic obstructive pulmonary disease (OR = 1.8), and diabetes (OR = 1.2).

REVIEW OF COMBINED SCORING SYSTEMS

The last category of scoring systems is a combination of the previous ones. Combined scoring systems incorporate elements from the three other risk adjustment categories to combine more information and potentially a better outcome prediction. The most notable of these scores is the Trauma and Injury Severity Score (TRISS) and A Severity Characterization of Trauma (ASCOT) score.

Boyd et al. introduced TRISS in 1987 by taking into account RTS, ISS, and age. TRISS calculates the estimated survival probability for a patient as:

$Ps = \frac{1}{1+e^{-b}}$, where $-b = b_0 + b_1(RTS) + b_2(ISS) + b_3(A)$

and $A = age\ indicator$ where 0 if ≤ 55 or 1 if > 55 years old.

The coefficients (b_{0-3}) are acquired from regression analysis on data from the Major Trauma Outcome Study. These coefficients differ for penetrating or blunt injury. If a patient is less than 15 years of age than the blunt mechanism formula is used in all cases.

Despite the advancement TRISS contributes by introducing combined scoring systems it has one major flaw – it relies too heavily on ISS. Therefore, without the prerequisite amount of variables TRISS fails to compute. Champion et al. in 1990 developed ASCOT to remedy this problem. ASCOT uses AIS anatomic injury scores to dodge ISS and its related short comings and outperformed TRISS in predictive ability. ASCOT however has yet to be widely accepted.

Combined scoring systems that incorporate both physiological and anatomical criteria create a more encompassing scoring metric. However, both the popular TRISS and ASCOT use age as a surrogate, which fails to deliver an accuate picture of trauma outcome for people with pre-existing medical conditions. The proposed CAT injury scoring system, presented in this thesis, remedies this problem by allowing for the inclusion of comborities in the foundation model that generates the score. The following chapters 3 and 4 present the methodology and results of the CAT injury scoring system.

Chapter Three: Research Methodology

This chapter describes the procedures taken to devise and test the CAT Scoring System. The rationale for this development and the corresponding limitations are discussed at the end. Data cleaning, variable manipulations, data investigation, and score creation and testing were all conducted using SAS ® 9.2 software. For reference, this SAS code can be found in Appendices A and B.

DESCRIPTION OF THE PROCEDURES

I. Data Cleaning & Variable Manipulations

All initial data were sourced from the National Trauma Data Bank Research Data Set Version 7.1. Although there was access to twenty-one different datasets with varying patient and facility information related to medical trauma, only six datasets were deemed necessary for this project. All datasets were obtained as DBF files or D-Base files and subsequently loaded directly into SAS. The selected NTDB datasets and the corresponding content descriptions can be found in Table 1.

Table 1

National Trauma Data Bank Datasets and Descriptions

Dataset	Description
RDS_COMORBID	Information pertaining to any pre-existing comorbid diseases the patient had upon arrival in the ED/hospital
RDS_DEMO	Includes information about the patient and incident demographics
RDS_DISCHARGE	Includes discharge and outcome information pertaining to the trauma incident
RDS_ED	Includes information pertaining events and measurements that take place in the ED
RDS_FACILITY	Includes information about the participating facilities
RDS_SCENE	Includes information pertaining to the scene of the trauma incident

*Datasets defined according to NTDB RDS 7.1 USER MANUAL – Table1: Data files and descriptions

RDS_COMORBID, RDS_DEMO, RDS_DISCHARGE, RDS_ED, RDS_FACILITY, and *RDS_SCENE* were imported into SAS and renamed *rdscomorbid, redsdemo, rdsdischarge, rdsed, rdsfacility,* and *rdsscene* respectively for convenience. This step is displayed in Appendix A – "Analysis Part 1" (pg. 115). The aim of this stage of the data analysis is to reconstruct these six datasets into one cohesive dataset containing extracted variables of interest and binary (present or absent) variables for each of the comorbidities. This is accomplished by sequentially merging these datasets by the incident key, *inc_key*, or facility key, *fac_key,* and manipulating the variable *prexcomorb* from *rdscomorbid*.

Overall there are approximately 88 variables provided across all six datasets; however, for the purpose of this project there is only interest in 39. A listing of these 39 variables, their descriptions, and category options or allowable ranges can be located in Table 2. Also a listing of each variable and its origin dataset can be found below in Table 3. This information was collected directly from the NTDB User Manual.

Table 2

Main Variable Definitions and Variable Category Restrictions

Variable Name	Variable Definition	Variable Options
acslevel	ACS verification level	"I"," II", "III", "IV" or "Not Applicable"
acspedlev	ACS verification level for pediatric hospital	I," II", "Not Applicable"
acs_edrts	Recalculated Revised Trauma Score In ED by ACS	Any number between 0 and 8
acs_ps	Recalculated TRISS Survival Probability by ACS.	Any real number between 0.00 and 1.00
age	The age of the patient on arrival to the hospital	0 – 89 years, -1 represents patient 89 – 120 years old
disstatus	Discharge Status	"Alive", "Dead"
edbasedef	Base Deficit/Excess In ED	Any integer between –80 and +80
edgcseye	Lowest Glasgow Eye Component In ED	Values for Adults (>5 yrs old): 1 = Does Not Open Eyes 2 = Opens Eyes to Pain 3 = Opens Eyes to Commands 4 = Spontaneous Eye Opening Values for Infants and Children: 1 = No Response 2 = Pain 3 = Verbal Stimuli 4 = Spontaneous
edgcsmotor	Lowest Glasgow Motor Component In ED	Values for Adults (>5 yrs old): 1 = None 2 = Extensor posturing in response to painful stimulation 3 = Flexor posturing in response to painful stimulation 4 = General withdrawal in response to painful stimulation 5 = Localization of painful stimulation 6 = Obeys commands with appropriate motor response Values for Infants and Children: 1 = None 2 = Abnormal flexion (decerebrate) 3 = Abnormal flexion (decerebrate) 4 = Withdraws to pain 5 = Withdraws to touch 6 = Normal Spontaneous Movement
edgcstotal	Glasgow Coma Scale Total In ED	Any integer between 3 and 15

*Variables defined according to NTDB RDS 7.1 USER MANUAL – Appendix A

Table 2 (Continued)

Main Variable Definitions and Variable Category Restrictions

Variable Name	Variable Definition	Variable Options
edgcsverb	Lowest Glasgow Verbal Component In ED	Values for Adults (>5 yrs old): 1 = None 2 = Incomprehensible words 3 = Inappropriate Words 4 = Confused 5 = Oriented Values for Child: 1 = No Response 2 = Incomprehensible sounds 3 = Inappropriate Cries 4 = Confused 5 = Oriented Values for Infant: 1 = No Response 2 = Moans to Pain 3 = Cries to Pain 4 = Irritable Cries 5 = Coos, Babbles.
edrtrs	Revised Trauma Score In ED	Any number between 0 and 8
edsysbp	The initial assessment in the ED of the systolic blood pressure	Any integer between 0 and 300
edtemp	First Temperature In ED	Any real number between 0 and 110
fac_key	Facility Key	
fimexpress	FIM Expression Score At Discharge	1 = Dependent-Total Help Required 2 = Dependent-Partial Help Required 3 = Independent with Device 4 = Independent 8 = Not Applicable (e.g., < 7 yrs. old or died) 9=Unknown
fimfeed	FIM Self-feeding Score At Discharge	1=Dependent-Total Help Required 2=Dependent-Partial Help Required 3=Independent with Device 4=Independent 8 = Not Applicable (e.g., < 7 yrs. old or died) 9=Unknown

*Variables defined according to NTDB RDS 7.1 USER MANUAL – Appendix A

Table 2 (Continued)

Main Variable Definitions and Variable Category Restrictions

Variable Name	Variable Definition	Variable Options
fimlocomt	FIM Locomotion Score At Discharge	1 = Dependent-Total Help Required 2 = Dependent-Partial Help Required 3 = Independent with Device 4 = Independent 8 = Not Applicable (e.g., < 7 yrs. old or died) 9=Unknown
fimtotal	Total FIM Score	Any integer between 1 and 12
gender	Gender	"Male", "Female"
ICUdays	Days Of Total Stay In ICU	Less than or equal to the Hospital Length of Stay and between 0 – 364 days
injurytype	Injury Type	"Blunt", "Burn", "Penetrating"
inc_key	Incident key	
iss	Total Injury Severity Score	An integer between 0 and 75
los	Length Of Stay In Hospital	Integer between 0 – 364 days
payment	Principal Payment Source	"Automobile Insurance", "Blue Cross/Blue Shield", "CHAMPUS", "Government/Military Insurance", "Liability Insurance/Under Litigation", "MCH and Crippled Children's", "Managed Care Organization", "Medicaid", "Medicare", "No Charge", "No Fault Insurance", "None", "Not Done/Not Doc", "Organ Donor Subsidy", "Other", "Other Commercial Indemnity Plan", "Pending", "Private Charity", "Self Pay", "Worker's Compensation"
prexcomor	Pre-existing comorbid factor present at the point of patient arrival in the ED	44 preselected comorbidities translated from ICD-9-CM codes
race	Race	"Asian or Pacific Islander", "Black", “Hispanic”, "Native American or Alaskan Native", "Other", "White, not of Hispanic Origin"
region	Geographic region for the hospital	"Midwest", "Northeast", "South", "West"

*Variables defined according to NTDB RDS 7.1 USER MANUAL – Appendix A

Table 2 (Continued)

Main Variable Definitions and Variable Category Restrictions

Variable Name	Variable Definition	Variable Options
sceneeye	Lowest Glasgow Eye Component At The Scene	Values for Adults (> 5 yrs old): 1 = None 2 = Pain 3 = Voice 4 = Spontaneous Values for Children and Infants: 1 = No Response 2 = Pain 3 = Verbal Stimuli 4 = Spontaneous
scenegcsaq	GCS Assessment Qualifier At The Scene	"L" = Initial GCS components at scene are legitimate values, without interventions such as intubation and sedation. "S" = Patient chemically sedated when initial GCS components assessed at scene. "T" = Patient intubated when GCS components assess at scene. "TP" = Patient intubated and chemically paralyzed when GCS components assessed at scene
scenemotor	Lowest Glasgow Motor Component At The Scene	Values for Adults (>5 yrs old): 1 = None 2 = Extensor posturing in response to painful stimulation 3 = Flexor posturing in response to painful stimulation 4 = General withdrawal in response to painful stimulation 5 = Localization of painful stimulation 6 = Obeys commands with appropriate motor response 9 = Not Done/Not Documented Values for Infants and Children: 1 = None 2 = Abnormal flexion (decerebrate) 3 = Abnormal flexion (decerebrate) 4 = Withdraws to pain 5 = Withdraws to touch 6 = Normal Spontaneous Movement 9 = Not Done/Not Documented
scenetotal	Glasgow Coma Scale Total At The Scene	Any integer between 3 and 15

*Variables defined according to NTDB RDS 7.1 USER MANUAL – Appendix A

Table 2 (Continued)

Main Variable Definitions and Variable Category Restrictions

Variable Name	Variable Definition	Variable Options
scenever	Lowest Glasgow Verbal Component At The Scene	Values for Adults (>5 yrs old): 1 = None 2 = Incomprehensible words 3 = Inappropriate Words 4 = Confused 5 = Oriented Values for Child: 1 = No Response 2 = Incomprehensible sounds 3 = Inappropriate Cries 4 = Confused 5 = Oriented Values for Infant: 1 = No Response 2 = Moans to Pain 3 = Cries to Pain 4 = Irritable Cries 5 = Coos, Babbles
statelevel	State Designation level	"I", "II", "III", "IV", "Other" or "Not Applicable"
tempscale	Temperature Scale	"C" = Celsius "F" = Fahrenheit
triss_prob	TRISS Survival Probability	Any number between 0.00 and 1.00
ventdays	Ventilator Support Days	Less than or equal to the Hospital Length of Stay and between 0 – 364 days
yoadmit	Year of First Recorded Patient's Arrival At Reporting Hospital ED	Greater than or equal to 1993

*Variables defined according to NTDB RDS 7.1 USER MANUAL – Appendix A

Table 3

National Trauma Data Bank Sources for Main Variables

Variable Name	Origin Dataset(s)
acslevel	RDS_FACILITY
acspedlev	RDS_FACILITY
acs_edrts	RDS_ED
acs_ps	RDS_ED
age	RDS_DEMO
disstatus	RDS_DISCHARGE
edbasedef	RDS_ED
edgcseye	RDS_ED
edgcsmotor	RDS_ED
edgcstotal	RDS_ED
edgcsverb	RDS_ED
edrtrs	RDS_ED
edsysbp	RDS_ED
edtemp	RDS_ED
fac_key	RDS_DEMO, RDS_DISCHARGE, RDS_ED, RDS_FACILITY, RDS_SCENE
fimexpress	RDS_DISCHARGE
fimfeed	RDS_DISCHARGE
fimlocomt	RDS_DISCHARGE
fimtotal	RDS_DISCHARGE
gender	RDS_DEMO
ICUdays	RDS_DISCHARGE
injurytype	RDS_SCENE
inc_key	RDS_COMORBID, RDS_DEMO, RDS_DISCHARGE, RDS_ED, RDS_SCENE
iss	RDS_ED
los	RDS_DISCHARGE
payment	RDS_DISCHARGE
prexcomor	RDS_COMORBID
race	RDS_DEMO
region	RDS_FACILITY
sceneeye	RDS_SCENE
scenegcsaq	RDS_SCENE
scenemotor	RDS_SCENE
scenetotal	RDS_SCENE
scenever	RDS_SCENE
statelevel	RDS_FACILITY
tempscale	RDS_ED
triss_prob	RDS_ED
ventdays	RDS_DISCHARGE
yoadmit	RDS_DEMO, RDS_ED

The first step in the processing of the NTDB datasets was to isolate the datasets to include only the facilities and years of ED admissions where pre-existing conditions were recorded. This was accomplished by merging *rdsed* (n=1926245) and *rdscomorbid* (n=602000) by *inc_key* to create a new dataset called *TempComorb* (n=2197342). *TempComorb* was then reduced to include only the variables *inc_key*, *fac_key*, *yoadmit*, and *prexcomor*. The reason there were more observations in *TempComorb* than initially expected from this merge is that some observations had several pre-existing conditions. Therefore a single inc_key might be listed several times, each with a different pre-existing condition. The comorbidities that are provided in *prexcomor* are listed in Table 4 along with their related ICD-9-CM codes, if provided, sourced directly from the NTDB User Manual.

Table 4

Pre-existing Comorbidity Factors and Related ICD-9-CM Codes

Comorbidity	ICD-9-CM Codes
Acquired Coagulopathy	286.7
Active Chemotherapy	V58.1
Alzheimer's Disease	290.0-290.13, 331.0
Asthma	493.0-493.9
Bilirubin > 2mg % (on Admission)	Code not provided
Chronic Alcohol Abuse	303.9
Chronic Dementia	290.10
Chronic Demyelinating Disease	341.0-341.9
Chronic Drug Abuse	304.0-304.9
Chronic Obstructive Pulmonary Disease	493.2, 496
Chronic Pulmonary Condition	496
Concurrent or Existence of Metastasis	Code not provided
Congestive Heart Failure	428.0, 425.0-425.9
Cor Pulmonale	416.8, 415.0, 416.9
Coronary Artery Disease	414.9, 414.0
Coumadin Therapy	Code not provided
CVA/Hemiparesis (Stroke with Residual)	342.0-342.9
Dialysis (Excludes Transplant Patients)	V56.0, V45.1, V56.8

*Variable codes defined according to NTDB RDS 7.1 USER MANUAL – Appendix E

Table 4 (Continued)

Pre-existing Comorbidity Factors and Related ICD-9-CM Codes

Comorbidity	ICD-9-CM Codes
Spinal Cord Injury	any 806, 952-954
Systemic Lupus Erythematous	710.0
Transplants	Comorbidity not listed
Undergoing Current Therapy	Code not provided
Documented History of Cirrhosis	571.2, 571.5
Documented Prior History of Pulmonary Disease with Ongoing Active Treatment	Code not provided
Gastric or Esophageal Varices	456.0-456.2
Hemophilia	286.0-286.4
History of Cardiac Surgery	V45.0, P35.00-39.99, V42.1, V42.2, V42.2, V43.3
History of Psychiatric Disorders	any V11, V40.2
HIV/AIDS	079.53
Hypertension	any 401, 402.00, 402.10, 402.90
Inflammatory Bowel Disease	558.9
Insulin Dependent	250 (5th digit assignment for each comorbid factor)
Multiple Sclerosis	340
Non-Insulin Dependent	250 (5th digit assignment for each comorbid factor)
Obesity	278.00-278.01
Organic Brain Syndrome	310.9
Parkinson’s Disease	332.0
Peptic Ulcer Disease	any 533
Pre-existing Anemia	285.0, 285.8, 285.9
Pregnancy	any V22
Rheumatoid Arthritis	714.0-714.9
Routine Steroid Use	Comorbidity not listed
Seizures	780.3
Serum Creatinine > 2 mg % (on Admission)	Code not provided

*Variable codes defined according to NTDB RDS 7.1 USER MANUAL – Appendix E

TempComorb was then used to create a new dataset called *Comorb* as seen in the SAS code, Appendix A - “Analysis Part 2” (pg. 117). In this dataset a comorbidity counter, *CombCount,* was developed to assist in the elimination of unwanted data. The variable *Personcomb* was then created to add the total number of *CombCount* for each inc_key, keeping only the final inc_key with the total comorbidity count. Lastly the variable *comorbidity* was devised to translate the comorbidity counts from *CombCount* into a binary indicator. Only the variables *inc_key*, *fac_key*, *yoadmit*, *Personcomb*, and *comorbidity* were kept reducing the dataset to 1926245 observations. These variables are defined as follows:

$$CombCount = \begin{cases} 0 & if \quad Prexcomor = "\ " \\ 1 & otherwise \end{cases}$$

$$Personcomb = \begin{cases} 1 & if \quad CombCount > 0 \\ 0 & otherwise \end{cases}$$

$$Comorbidity = \begin{cases} 1 & if \quad Personcomb > 0 \; in \; last \; inc\_key \\ 0 & otherwise \end{cases}$$

The dataset *Comorb* was then sorted by *fac_key* and *yoadmit*, and the mean and count for these two variables by *Comorbidity* was taken and outputted into a dataset called *Meanout* (n=2199). *Meanout* was then reset to only include records where the mean is greater than 0, eliminating any facility that does not record comorbidities or any years of ED admission in which there are no recorded comorbidities for any patient. This reduced *Meanout* to an n=1125. The variable *_FREQ_*, which is automatically generated from *proc means,* was summed to determine a final observation count of 12050549. This count will be an important indicator to check for the accuracy in merging the other NTDB datasets later on. Example 1 and Example 2 below illustrate the data transition from the initial NTDB dataset merging into *TempComorb* and later in its final reduced form for four of the facilities.

Example 1

Data Format after the Initial Merging of NTDB Datasets rdsed and rdscomorbid

Facility Key (*fac_key*)	**Year of ED Admission (*yoadmit*)**	**Pre-existing Conditions Provided (*prexcomor*)**
1	2002 2003 2004 2005 2006	X X X X
2	2002 2003 2004 2005 2006	X X X X X
6	2002 2003 2004 2005 2006	X
19	2002 2003 2004 2005 2006	

Example 2

Data Format after the Reduction to only include Facilities and ED Years that Provide Comorbidities

Facility Key (*fac_key*)	**Year of ED Admission (*yoadmit*)**	**Pre-existing Conditions Provided (*prexcomor*)**
1	2002 2003 2004 2005	X X X X
2	2002 2003 2004 2005 2006	X X X X X
6	2002	X

Now that only the desired facilities and ED admission years for those facilities are isolated, the NTDB datasets can be merged into one final workable dataset. The merges occurred in the following order:

1. *Meanout* was merged with *rdsed* by *fac_key* and *yoadmit* if in *Meanout* creating dataset *M1*
2. *M1* was merged with *redsdemo* by *inc_key* if in *M1* creating dataset *M2*
3. *M2* was merged with *rdsdischarge* by *inc_key* if in *M2* creating dataset *M3*
4. *M3* was merged with *rdsscene* by *inc_key* if in *M3* creating *M4*
5. *M4* was merged with *rdsfacility* by *fac_key* if in *M4* creating *M5*

Checking for accuracy, all merged datasets had an n=12050549. Before the last NTDB *rdscomorbid* could be merged to create the final dataset it needed to be altered so that each comorbidity category had its own variable, in binary form as present or absent, for each patient. This will be necessary to perform the regression techniques used in the latter analysis stage. The SAS code used to accomplish this is displayed in Appendix A under "Analysis Part 3" (pg. 119).

To reformat the comorbidity variable *prexcomor*, *rdscomorbid* was set to a new dataset called *Bincomor* (n=602000). Each comorbidity category was then assigned a binary variable as described below:

$$acqcoag = \begin{cases} 1 & if \quad Prexcomor = \text{"}Acquired\ coagulopathy\text{"} \\ 0 & otherwise \end{cases}$$

$$activechemo = \begin{cases} 1 & if \quad Prexcomor = \text{"}Active\ Chemotherapy\text{"} \\ 0 & otherwise \end{cases}$$

$$alzhiemers = \begin{cases} 1 & if \quad Prexcomor = \text{"}Alzhiemers\ Disease\text{"} \\ 0 & otherwise \end{cases}$$

$$\begin{matrix} \vdots & \vdots & \vdots \end{matrix}$$

$$undergoingthpy = \begin{cases} 1 & if \quad Prexcomor = "Undergoing\ Current\ Therapy\ " \\ 0 & otherwise \end{cases}$$

The format of a patient's comorbidity data at this stage of the comorbidity manipulation can be seen below in Example 3. This patient and *inc_key* are fictitious and used for illustrated purposes only.

Example 3

Step 1 of the Creation of Horizontal, Single Record, Binary Comorbidity Variables

inc_key	cirrhosis	alcoholabuse	drugabuse	pregnancy
777777	0	0	0	1
777777	1	0	1	0
777777	1	0	0	0
777777	0	0	0	0

*****For illustrative purposes only – patient data is fictitious

After sorting by inc_key, *Bincomorb* was reassigned to the dataset *Attempt2*. In *Attempt2* new comorbidity variables were drafted for each comorbidity category. These variables represent the sum of the previously created comorbidity binary variables for each *inc_key*. An *inc_key* may be repeated for several rows of observations due to some patients possessing more than one pre-existing condition. These new variables are a running total of said conditions until the last repeated *inc_key*. This is shown in the continued fictitious data below in Example 4.

Example 4

Step 2 of the Creation of Horizontal, Single Record, Binary Comorbidity Variables

inc_key	cirrsum	alchsum	drugsum	pregsum
777777	0	0	0	1
777777	1	0	1	1
777777	2	0	1	1
777777	2	0	1	1

*****For illustrative purposes only – patient data is fictitious

A listing of all the new totaled comorbidity variables can be seen in Table 5. These variables will be referenced frequently throughout the rest of this paper.

Table 5

Pre-existing Comorbidity Variables and Descriptions

Variable Name	Variable Description
acqsum	Acquired Coagulopathy
alchsum	Chronic Alcohol Abuse
alzhsum	Alzheimers Disease
anemsum	Pre-existing Anemia
arthsum	Rheumatoid Arthritis
ashsum	Asthma
bilsum	Bilirubin > 2mg % (on Admission)
bowelsum	Inflammatory Bowel Disease
brainsum	Organic Brain Syndrome
chemosum	Active Chemotherapy
cirrsum	Documented History of Cirrhosis
corartsum	Coronary Artery Disease
corpsum	Cor Pulmonale
coudsum	Coumadin Therapy
cvasum	CVA/Hemiparesis (Stroke with Residual)
demsum	Chronic Dementia
dialsum	Dialysis (Excludes Transplant Patients)
dmysum	Chronic Demyelinating Disease
drugsum	Chronic Drug Abuse
gasvsum	Gastric or Esophageal Varices
heartsum	Congestive Heart Failure
hemosum	Hemophilia
hivsum	HIV/AIDS
hpytsum	Hypertension
htycardsum	History of Cardiac Surgery
insldepsum	Insulin Dependent
lupussum	Systemic Lupus Erythematous
metasum	Concurrent or Existence of Metastasis
msclsum	Multiple Sclerosis
ninsldepsum	Non-Insulin Dependent
obplsum	Chronic Obstructive Pulmonary Disease
obstsum	Obesity
parksum	Parkinson’s Disease
pepusum	Peptic Ulcer Disease
ppulsum	Documented Prior History of Pulmonary Disease with Ongoing Active Treatment
pregsum	Pregnancy
psychsum	History of Psychiatric Disorders
pulcsum	Chronic Pulmonary Condition
seizsum	Seizures
sercsum	Serum Creatinine > 2 mg % (on Admission)

Table 5 (Continued)

Pre-existing Comorbidity Variables and Descriptions

Variable Name	Variable Description
spinalsum	Spinal Cord Injury
sterdsum	Routine Steroid Use
thpysum	Undergoing Current Therapy
transpsum	Transplants
prescomorb	Presence of any of the above comorbidities

Attempt2 was then set to the new dataset *bincomorb2* and only the last observation for each *inc_key* was kept. In the case where there was only one observation per *inc_key* then that observation was retained. This reduced the data from 602000 to 330903 observations. This reduction can be seen in the continued example data in Example 5.

Example 5

Step 3 of the Creation of Horizontal, Single Record, Binary Comorbidity Variables

inc_key	cirrsum	alchsum	drugsum	pregsum
777777	2	0	1	1

*For illustrative purposes only – patient data is fictitious

After the reduction, *Bincomorb2* was merged with *M5*, from the series of previous merges, by inc_key if it was in *M5* creating dataset *M6*. This step joined the altered *rdscomorbid* to the other five original NTDB datasets. Even though a combined dataset with all desired NTDB source datasets was developed, there was still additional data cleaning to be performed. *M6* was set to the dataset *ALL* where a binary variable for *disstatus* termed *dead* was created as follows:

$$dead = \begin{cases} 1 & if \quad disstatus = "dead' \\ 0 & otherwise \end{cases}$$

To correct for the reporting error of the same comorbidity category being listed more than once for a single inc_key, the comorbidity variables were redefined in the dataset *ALL* as follows:

$$acqsum = \begin{cases} 1 & if \quad acqsum > 1 \\ 0 & if \; acqsum = . \end{cases}$$

$$alchsum = \begin{cases} 1 & if \quad alchsum > 1 \\ 0 & if \; alchsum = . \end{cases}$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$transpsum = \begin{cases} 1 & if \quad transpsum > 1 \\ 0 & if \; transpsum = . \end{cases}$$

The changes to the binary comorbidity variables as reflected in the dataset *ALL* can be seen in the continued example below (Example 6).

Example 6

Step 4 of the Creation of Horizontal, Single Record, Binary Comorbidity Variables

inc_key	**cirrsum**	**alchsum**	**drugsum**	**pregsum**
777777	1	0	1	1

***For illustrative purposes only – patient data is fictitious**

The dataset *All* containing all six source datasets was assigned to a new dataset *Comorb.final*. Fortunately, the resulting dataset has 1250549 observations ensuring accurate data manipulation. A last comorbidity variable was created to represent the presence or absent of any pre-existing conditions by *inc_key*. This was accomplished by first creating a variable termed *comorbsum* that totaled all the comorbidity category binary variables by each inc_key. Then the general comorbidity binary variable, *prescomorb* was assigned as follows:

$$prescomorb = \begin{cases} 1 & if \quad comorbsum > 1 \\ 0 & otherwise \end{cases}$$

The resulting data format at this stage with the dataset *Comorb.final* can be seen in the continued example below (Example 7). Additional fictitious *inc_key* listings were included for varied data option illustration.

Example 7

Step 5 of the Creation of Horizontal, Single Record, Binary Comorbidity Variables

inc_key	cirrsum	alchsum	drugsum	pregsum	comorbsum	prescomorb
777777	1	0	1	1	3	1
888888	0	0	0	1	1	1
999999	1	1	0	0	2	1
111111	0	0	0	0	0	0

*****For illustrative purposes only – patient data is fictitious

The final stage of data cleaning and variable manipulation for this project was to ensure that all the variables of interest meet the NTDB User Manual variable category restrictions as explained in *Table 2*. Each of the 39 variables was examined individually and changes were made as necessary. The SAS code used for this stage is displayed in Appendix A under “Analysis Part 4” (pg. 128). “N/A” was changed to “Not Applicable” for consistency for the variable *acspedlev* and “D” was changed to missing for *gender*. Two observations less than -80 for *edbasedef* were set to missing. The variables *edgcstotal*, *Fimtotal*, *scenetotal* were set to be defined as:

edgcstotal = *edgcseye* + *edgcsmotor* + *edgcsverb*; where $3 < edgcstotal < 15$

Fimtotal = *Fimexpress* + *Fimfeed* + *Fimlocomt*; where $1 < Fimtotal < 12$

scenetotal = *sceneeye* + *scenemotor* + *scenever*; where $3 < scenetotal < 15$.

If any of the three composite variables were missing or if total variable was out of range then the total variable was set to missing. If *edrtrs, Fimexpress, Fimfeed, Fimlocomt, scenemotor,* or *scenever* equaled 9 for unknown then the variable was set to missing before creating total variables. If *Fimexpress, Fimfeed*, or *Fimlocomt* were equaled to 8 or "Not Applicable" the variables were set to 0 before creating *Fimtotal*. In the cases where *scenegcsaq* were "T" or "TP" for "patient intubated" scenever was set to missing.

Systolic Blood pressure variable *edsysbp* was set to missing if greater than 300 or less than 0. *Iss* was set to missing if any observations were greater than 75 or less than 0. If *tempscale* was marked as "C" for Celsius then *edtemp* was converted to Fahrenheit by the following, setting out of range values to missing:

$$edtemp = (edtemp * 1.8) + 32; \qquad \text{where } 0 < edtemp < 110 .$$

Length of stay variable, *Los*, and the number of ICU and ventilation days are dependent. Any variation from the following range was set to missing:

$$0 \leq Ventdays \leq ICUdays \leq Los \leq 364.$$

And lastly *age* was divided into the following four age categories to be used in future demographic analysis:

$$agecat = \begin{cases} 1 & if\ \ 0 < age < 55 \\ 2 & if\ 54 < age < 65 \\ 3 & if\ 64 < age < 75 \\ 4 & if\ age > 74\ or\ age = -1 \\ 0 & otherwise \end{cases}$$

After age categories were defined if *age* was less than 0 then *age* was set to missing. If a patient had an age category greater than 1 and pregnancy was indicated as *pregsum*=1 or present then *pregsum* was changed to indicate absence of condition or 0.

All of the data cleaning and variable manipulations were stored in the final dataset, *Comorb.cleanedfinal*. A reconstructed, cohesive dataset containing the extracted variables of interest and binary, present or absent, variables for each of the comorbidity categories was accomplished. Missing and out of range values were examined and corrected if necessary. At this stage, the data were primed for analysis. The data investigation techniques and score building methodology follow.

II. Methods for Data Investigation: Descriptive Statistics

Exploratory data analysis was conducted prior to regression analysis and injury score creation. The statistical techniques and programming methods to perform this part of the data investigation will be discussed in this section. The SAS code used to obtain the descriptive statistics is displayed in Appendix A under "Analysis Part 5" (pg. 129).

The descriptive statistics portion was divided into three groupings for analysis – categorical main variables, numerical main variables, and comorbidity variables. Main variables were also subsequently divided into outcome or predictor groups. *Los*, *ICUdays*, *Ventdays*, *fimexpress*, *fimfeed*, *fimlocomt*, and *fimtotal* are outcome variables. All other main variables are predictor variables. The SAS technique *proc means* was used for numerical main variables and *proc freq* was used for both categorical main variables and comorbidity variables.

Summary statistics were collected and organized into tables in the following manner:

1. Missing counts and percentages for all main variables

   SAS Technique: *proc means* with *n* and *nmiss* options for numerical variables - *nmiss* was divided by *n* for percentages

   proc freq for categorical variables – frequency missing was divided by total cumulative frequency plus frequency missing from the output for percentages

   Statistical Tests: none

2. Categorical variable counts and percentages by the present or absence of comorbidities or comorbidity status

   SAS Technique: *proc freq* using the variable *prescomorb* multiplied by the categorical variables in the *table* statement

   chisq was specified at the end of the *table* statement

   Statistical Tests: Chi-Square test to test for differences between comorbidity status

3. Numerical variable ranges, means, and standard deviations by comorbidity status

   SAS Technique: *proc means* with options *min*, *max*, *mean*, *std* and using the *by* statement for the variable *prescomorb*

   proc ttest with *prescomorb* in the *class* statement

   Statistical Tests: t-test to test for the equality of the means between patients with comorbidities and those without

4. Categorical outcome variable counts and percentages based on age category

   SAS Technique: *proc freq* using the variable *agecat* multiplied by the categorical variables in the *table* statement

   chisq was specified at the end of the *table* statement

   Statistical Tests: Chi-Square test to test for differences between age categories

5. Numerical outcome variable means and standard deviations based on age category

   SAS Technique: *proc means* with options *mean*, *std* and using the *by* statement for the variable *agecat*

   proc ttest with *agecat* in the *class* statement

   Statistical Tests: t-test to test for the equality of the means between patients of different age categories

6. Comorbidity variable counts and percentages amongst patients with comorbidities and all patients

   SAS Technique: *proc means* with option *n* using *prescomorb* in the *by* statement for within comorbidity counts and without the *by* statement for total counts

   Statistical Tests: none

7. Comorbidity variable counts and percentages based on age category

   SAS Technique: *proc freq* using the variable *agecat* multiplied by the comorbidity variables in the *table* statement

   chisq was specified at the end of the *table* statement

   Statistical Tests: Chi-Square test to test for differences between age categories

III. Methods for Data Investigation: Regression Analysis

After the exploratory data analysis was conducted and descriptive statistics were collected, model building and regression analysis were performed. The ultimate goal of this project was to explore the impact that comorbidities have on health outcomes in a trauma setting. Therefore developing and testing models with the incremental inclusion of comorbidities, as well as the standard predictors like age and TRISS were crucial to the development of an injury scoring system. This section relays the steps taken to examine the comorbidity relationship on trauma health outcomes.

The regression analysis was divided into four sections based on the type of health trauma outcome variable being measured. Individual models and analysis reports were created using this organizational division. The health outcome categories are death (*dead*), hospital length of stay (*Los*), the number of ICU days (*ICUdays*), and the number of ventilation days (*Ventdays*). The section is presented in this ordering.

- Death Outcome -

The health outcome most interest for this project is death because ultimately the developed injury scoring system will measure a patient's mortality risk. It is therefore critical that the models in this section are tested carefully. Three models were created, each adding in a new predictor working up to final model with comorbidity, TRISS, and age.

Model 1, as seen below, is the unadjusted model because it is simply testing the effect of comorbidities on patient mortality. It is shown to have *prescomorb* as the comorbidity variable but, in fact, can be interchanged for any of the individual comorbidity variables. *Prescomorb* indicates whether a patient has any pre-existing conditions present or not and is used as an example comorbidity predictor. When analysis was conducted all 44 individual binary comorbidity variables and *prescomorb* were tested separately in the model and reported. The variable *dead* in the model is a binary variable represented as 1 for dead and 0 for alive. Since the outcome is binary, logistic regression was chosen as the best means for analysis. The SAS procedure *proc logistic* was used and parameter estimates were outputted to report odds ratios and confidence intervals for the effect of pre-existing conditions on mortality risk. The SAS code for the regression analysis based on the outcome death can be found in Appendix A under "Analysis – Part 6" (pg. 131).

Model 1: Unadjusted Model

Unadjusted Logistic Regression Mortality Model:

$$\text{logit } P(\textit{Dead}) = \beta_0 + \beta_1 \textit{Prescomorb}$$

To evaluate if the addition of TRISS survival probability contributes any predictive effect beyond that of comorbidity alone, the variable *TRISS_prob* was included as a second predictor to the unadjusted model. This new model is depicted below as Model 2. To determine if age provided additional predictive effect to comorbidity and TRISS, *age* was added as a third predictor. It is of interest to know whether prescomorb adds predicitive ability to a model containging age. This is shown in Model 3 below. Please note that Model 3 is sometimes reference in later reported tables as the "binary model" because of its binary outcome. As in Model 1, *prescomorb* is a placeholder variable for any of the binary comorbidity variables and all 44 individual comorbidities were tested for each model.

Model 2

Logistic Regression Mortality Model with TRISS:

$$\text{logit } P(\textit{Dead}) = \beta_0 + \beta_1 \textit{Prescomorb} + \beta_2 \textit{TRISS\_prob}$$

Model 3: Binary Model

Logistic Regression Mortality Model with TRISS and Age:

$$\text{logit } P(\textit{Dead}) = \beta_0 + \beta_1 \textit{Prescomorb} + \beta_2 \textit{TRISS\_prob} + \beta_3 \textit{Age}$$

Two additional precautions were taken to obtain the most accurate results possible in the logistic regression analysis for Model 2 and Model 3. First multiple imputation was used to include the maximum amount of useable data in the logistic regression procedure. Secondly Net Reclassification Improvement (NRI) and Integrated Discrimination Index (IDI) techniques were employed to determine if a new model is significant compared to the old model.

Multiple imputation is a Monte Carlo statistical technique where missing variable information is replaced by simulated covariance matrix “placeholders”. These simulated versions are calculated or imputed several times then averaged to create one record for each piece of missing data. Multiple imputation is a better alternative to removing rows of data because one variable or variables in that row are missing (listwise deletion). This method is wasteful because the remaining data in that observation could be used or relevant. Instead multiple imputation uses other variables with low missing values to estimate the missing one. In our case those variables are *Los, ICUdays, Ventdays, edgcstotal, edrtrs,* and *iss*. The number of imputations needed depends on the rate of missing data in the variable being estimated. Often only a small number are needed which is fortunate because it has the potential of being computer resource heavy with large datasets. Imputation efficiency is computed by the following equation (Rubin, 1987):

$imputation\ efficiency = (1 + \frac{\gamma}{m})^{-1}$, where γ = missing rate, *m* = # of imputations.

The number of imputations used in this project was five which varies the efficiency rate from 0.85 for a 0.9 rate of missing to greater than 0.98 for a missing rate of less than 0.1.

In SAS, *proc mi* was used with a *nimpute* of 5 specifying the low missing variables and model variables in the *var* statement. *Proc logistic* was then implemented by *_imputation_* outputting the parameter estimates and covariance matrices. And lastly, *proc mianalyze* was used specifying stacking for the variance effect. The outputted parameter estimates from *proc mianalyze* were then used to compute odds ratios and confidence intervals for mortality risk.

Due to the large nature of the NTDB dataset, relying on traditional tests, confidence intervals, and p-values proves unfruitful. Incremental changes in improvement between models need to be measured to gather an accurate picture of the predicator variable’s added effect. For that reason, NRI and IDI techniques were employed. NRI and IDI are better measures than just measuring increases in the Area under the ROC curve as this technique may introduce issues with calibration. Also, the Area under the ROC curve can worsen with the addition of several variables. NRI is calculated by dividing patients into those with the event (or death) and those who are a non-event (or alive). Patients in the event grouping are assigned 1 for upward movement, -1 for downward movement, and 0 for no movement. The opposite is assigned to those patients in the alive group. Each group’s reclassification number assignments are summed and divided by the number of patients in each group. NRI is the difference in these quantities. NRI can be interpreted as the net gain or loss in the reclassification proportion. The NRI equation is as follows (Pencina, 2008):

$$NRI = [P(up|D=1) – P(down|D=1)] – [P(up|D=0) – P(down|D=0)].$$

Unlike NRI, IDI uses probability differences instead of event and non-event categories. IDI can also be seen as the “difference between improvement in average sensitivity and any potential increase in average ‘one minus specificity’.” (Pencina, pg.163). IDI is defined as follows:

$$IDI = (IS_{new} – IS_{old}) – (IP_{new} – IP_{old}),$$

where IS equals the integral of sensitivity over cut-off values from (0, 1) and IP equals the integral of ‘one minus specificity’ for the new and old models.

NRI and IDI were computed in SAS using the *rocplus* macro specifying the old variable as *triss_prob* for Model 2 or *triss_prob* and *age* for Model 3. The comorbidity variables were individually set to the added variable option and the id variable was given as *inc_key*. The risk cut-offs were indicated as 0.01, 0.02, 0.03, and 0.05. This macro was created by Eric Bergstralh in 2008 after the release of the Pencina paper. The *rocplus* macro is included in its entirety for reference in Appendix B.

- Length of Stay, Number of ICU Days, and Number of Ventilation Days Outcomes

Unlike the health outcome death, the outcomes length of hospital stay, number of

ICU days, and number of ventilation days are not binary but counts. In this case, the final injury scoring model is not built off of the findings from these outcomes. However, such an analysis would provide insight into the effect that pre-existing conditions have on quality of life and care factors in trauma.

To examine the impact that comorbidities have on length of stay, number of ICU days, and number of ventilation days, means and standard deviations for the number of days were collected for patients with pre-existing conditions and for those without. This step was done for all 44 binary comorbidities and for the general *prescomorb*. This was accomplished in SAS using the *proc means* procedure with the comorbidity variable in a *class* statement and the health outcome variable in the *var* statement. The SAS code for the regression analysis based on these outcomes can be found in Appendix A under "Analysis – Part 7" (pg. 221).

In addition to outcome means, the adjusted comorbidity effect was also of interest. Models were built for each of the three health outcomes including TRISS survival probability and comorbidity variables as predictors. In accordance with the mortality models, prescomorb may be replaced by any one of the 44 binary comorbidity variables. Model 4, Model 5, and Model 6 represent the outcomes *los*, *ICUdays*, and *Ventdays* respectively and are shown below. Adjusted comorbidity effects and related confidence intervals were computed in SAS using *proc reg* with the *clb* option.

Model 4

Regression Length of Hospital Stay Model:

$$\text{LOS} = \beta_0 + \beta_1 \textit{TRISS\_prob} + \beta_2 \textit{Prescomorb}$$

Model 5

Regression Number of ICU Days Model:

$$\text{ICUdays} = \beta_0 + \beta_1 \textit{TRISS\_prob} + \beta_2 \textit{Prescomorb}$$

Model 6

Regression Number of Ventilation Days Model:

$$\text{Ventdays} = \beta_0 + \beta_1 \textit{TRISS\_prob} + \beta_2 \textit{Prescomorb}$$

IV. CAT Severity Score Creation and Testing

Data cleaning, variable manipulation, exploratory data analysis, and regression analysis primed the NTDB data for the creation of a new severity score. It is at this stage of the project where the work shifts from analysis to development. This section describes the procedures taken to create the Comorbidity-Age-TRISS (CAT) injury scoring system. For reference, the SAS code used to complete this stage can be found in Appendix A "Analysis - Part 8" (pg 240).

As inferred by its name, the CAT injury scoring system was extracted from the analysis performed with the full, three predictor variable mortality model comprising of *prescomorb*, *TRISS_prob*, and *age* (Model 3). For the purposes of this section, Model 3 will be referred to as the binary model. The odds ratios that resulted from conducting a logistic regression analysis on the binary model were used to replace the present option in the binary comorbidity variables and these variables were subsequently renamed. An example of this formatting change using the pre-existing conditions asthma and cirrhosis is displayed below:

$$ashsum = \begin{cases} 1 & if \quad asthma\,present \\ 0 & otherwise \end{cases} \rightarrow \quad ashcomp = \begin{cases} 0.488 & if \quad asthma\,present \\ 0 & otherwise \end{cases}$$

$$cirrsum = \begin{cases} 1 & if\ cirrhosis\,present \\ 0 & otherwise \end{cases} \rightarrow \quad cirrcomp = \begin{cases} 3.941 & if\ cirrhosis\,present \\ 0 & otherwise \end{cases}$$

In order to create a general comorbidity variable (*comorscore*), indicting the presence or absence of any pre-existing conditions, all patient individual conditions were multiplied. For example, if a patient had both cirrhosis and asthma but no other pre-existing conditions then their *comorscore* would be 3.941 * 0.488 or 1.923. This CAT injury score relates that this patient has a 92.3% higher risk of mortality than a patient with no pre-existing conditions.

An additional alternative approach was taken in the creation of the CAT injury scoring system. Instead of changing all of the binary comorbidity variables to their corresponding mortality odds ratios from the binary model, only the comorbidities where their odds ratios deemed it harmful were changed. All beneficial comorbidities, or comorbidities whose odds ratios are 1 or less than 1, were assigned 1. Using the same example patient from above, the comorbidity variables are changed as follows:

$$ashsum = \begin{cases} 1 & if \quad asthma\,present \\ 0 & otherwise \end{cases} \rightarrow \quad ashcomp = \begin{cases} 1 & if \quad asthma\,present \\ 0 & otherwise \end{cases}$$

$$cirrsum = \begin{cases} 1 & if\ cirrhosis\,present \\ 0 & otherwise \end{cases} \rightarrow \quad cirrcomp = \begin{cases} 3.941 & if\ cirrhosis\,present \\ 0 & otherwise \end{cases}$$

A new general comorbidity, *comorscore2*, was created in the same way as *comorscore* using the new comorbidity variable odds ratios. Continuing with the same example as before, the patient's CAT injury score would now be 1 * 3.941 or 3.941 and now the patient's mortality risk is 394.1% greater than that of a patient with no pre-existing conditions.

To set up a performance comparison between the two new versions of the CAT injury scoring system and the old binary model, logistic regressions were performed in the same manner as with the mortality binary model substituting *comorscore* and *comorscore2* for *prescomorb*. Odds ratios, confidence intervals, NRI, IDI, and the corresponding p-values were obtained through this method and compared to the old binary model values. The models used in the logistic regression can be found below in Model 7 and Model 8.

Model 7: Score Model (1)

Logistic Regression Mortality Model with Unaltered CAT Scoring Variable:

$$\text{logit } P(\textit{Dead}) = \beta_0 + \beta_1 \textit{Comorscore} + \beta_2 \textit{TRISS\_prob} + \beta_3 \textit{Age}$$

Model 8: Score Model (2)

Logistic Regression Mortality Model with Beneficial CAT Scoring Variable:

$$\text{logit } P(\textit{Dead}) = \beta_0 + \beta_1 \textit{Comorscore2} + \beta_2 \textit{TRISS\_prob} + \beta_3 \textit{Age}$$

For comparison purposes, *comorscore* and *comorscore2* were substituted for *prescomorb* into the models for the regression analysis for length of hospital stay, number of ICU days, and number of ventilation days. Analysis was performed similarly and adjusted comorbidity effects and corresponding confidence intervals were obtained. The models for these procedures are can be found below (Model 9 – Model 14).

Model 9:

Regression Length of Hospital Stay Model with Unaltered CAT Scoring Variable:

$$LOS = \beta_0 + \beta_1 \textit{TRISS\_prob} + \beta_2 \textit{Comorscore}$$

Model 10:

Regression Number of ICU Days Model with Unaltered CAT Scoring Variable:

$$ICUdays = \beta_0 + \beta_1 \textit{TRISS\_prob} + \beta_2 \textit{Comorscore}$$

Model 11:

Regression Number of Ventilation Days Model with Unaltered CAT Scoring Variable:

$$Ventdays = \beta_0 + \beta_1 \textit{TRISS\_prob} + \beta_2 \textit{Comorscore}$$

Model 12:

Regression Length of Hospital Stay Model with Beneficial CAT Scoring Variable:

$$LOS = \beta_0 + \beta_1 \textit{TRISS\_prob} + \beta_2 \textit{Comorscore2}$$

Model 13:

Regression Number of ICU Days Model with Beneficial CAT Scoring Variable:

$$ICUdays = \beta_0 + \beta_1 TRISS\_prob + \beta_2 Comorscore2$$

Model 14:

Regression Number of Ventilation Days Model with Beneficial CAT Scoring Variable:

$$Ventdays = \beta_0 + \beta_1 TRISS\_prob + \beta_2 Comorscore2$$

RATIONALE FOR SOLUTION CHOICE

The full binary mortality model was deemed the best model to derive the CAT injury scoring system because of its inclusion of the three variables of interest. Using a model with both comorbidities and age enhanced the potential for accurate mortality prediction. Previous research suggests that combined scoring systems lend more credibility to an estimation.

Ultimately two CAT injury scoring systems were devised – the unaltered CAT scoring system and the beneficial CAT scoring system. It requires more biological research into these comorbidities and how it affects the hospital's treatment of individuals with these pre-existing conditions. The rationale behind this dual approach will be discussed further in the results chapter.

LIMITATIONS

The most prominent limitations in the methodology for the development of the CAT injury scoring system all trace back to the origin of the data. As with any social information, not collected in a lab under controlled conditions, this data has intrinsic flaws. Honest attempts were made to correct, alleviate, or report these issues in the data analysis process. The NTDB data limitations can be summarized into the follow categories:

V. Data Quality

Due to NTDB data collection process, data quality and completeness may vary between participating hospitals and facilities. Even though NTDB makes attempts to regulate and modify the incoming data from trauma registries, differences in data quality are inevitable. For this reason, analysis conducted using the aggregate data is subject to a degree of interpretation and error.

VI. Lack of Population Data

The six source NTDB datasets are collected from convenience samples or from hospitals and facilities that are willing and able to participate. This automatically skews the demographic information to those collected from large, more handsomely funded hospitals. It also might inflate the reported injuries that are considered to be severe as these are more likely to be documented.

VII. Selection and Information Bias

NTDB sample data naturally leads to selection bias because it is a convenience

sample. This occurs when the inclusion or exclusion criteria of a hospital varies for a given condition or definition causing misleading analysis results. A known example of this in the NTDB source data is how hospitals classify patients as Dead On Arrival (DOA). Some hospitals exclude these cases altogether while others report them. Information bias also occurs due to varying hospital definitions for reporting.
There is a level of subjectivity to how each doctor or clinician rates a measure of patient functioning.

VIII. Missing Data

Lastly, missing data is almost always a hurdle to any social or medical data analysis and the NTDB data is no exception. The variables of interest varied in the level of missingness. Multiple imputation was used to tried to alleviate some of this potential bias, relying on the variables with more complete profiles to fill in the gaps as best as possible.

Chapter Four: Results

The following chapter describes the results from the procedures taken to develop the CAT injury scoring system. The results are organized into three categories: descriptive statistics, regression analysis, and CAT severity score testing. The chapter is then concluded by a short discussion on the unexpected problems and findings encountered in the research.

DESCRIPTION OF THE OUTCOME RESULTS

I. Results of Data Investigation: Descriptive Statistics

Tables 6 – 12 summarize the information obtained from the descriptive statistics stage of the methodology described in Appendix A, "Analysis Part 5" (pg 129). Summary Statistics table descriptions and pertinent findings are as follows:

Table 6: (pg. 58)

Description:

Table 6 gives the missing data count and percentage for each main variable of interest from the original NTDB datasets. The table also indentifies each variable as a predictor or outcome.

Findings:

The amount of missing data varies from variable to variable. Some variables like gender (0.60%) or the year of hospital admission (0.00%) have little to no missing data, but other variables like *edbasedef* have a very high level of missing information (84.62%). It is important to note the presence of missing information and its degree, particularly in the variables used in model creation, to modify statistical approaches for accuracy. The use of MI rectifies issues that occur due to missingness.

Table 7: (pg. 59)

Description:

Table 7 displays the counts and percentages for the categorical main variables by comorbidity status. Percentages are given in three ways: the percentage of the overall variable data (%total), the percentage within the variable comorbidity category (%w/in), and the percentage within the variable category (%across). As an example, for the group of patients in ACS verification Level I with no comorbidities, they comprise 28.16% of all the ACS verification levels, 38.53% of all ACS verification levels without comorbidities, and 77.82% of ACS verification Level I.

Findings:

Since *yoadmit* had no missing data, it can be seen that overall 26.46% or 330903 of the patients were reported as having pre-existing conditions. The year 2004 reported the highest amount of patients with comorbidities, while 2007 reported the highest number of patients without. With the exception of 2002, all years reported about 21% of the data.

87.54% of patients were reported as having been injured in result from a blunt trauma and the majority of patients were male (64.71%). 32.10% of female patients were reported as having pre-existing conditions compared to 23.46% of the males. A typical patient would most likely be White from Non-Hispanic origin (67.66%) and from the South (36.63%). The Northeast reported the highest percentage of patients with comorbidities (38.01%). The majority of the payment options stated the greatest number of patients in the no comorbidities category; however, Medicare reported 55.22% of its patients with comorbidities.

Most of the data for both ACS verification levels and pediatric ACS verification levels are indicated as not applicable, yet the majority of patients fell in the state designation level I. For *disstatus*, 5.98% of patients noted as having pre-existing conditions died. This was a clinically meaningful difference in death percentage from patients who were reported with no pre-existing conditions (4.28%).

Glasgow eye, motor, and verbal components in the ED appear to have equally proportioned percentages of patients within each comorbidity category, per variable category. Most patients fell in the fully normal range of spontaneous eye opening, appropriate motor response to commands, and oriented. However slightly more patients with comorbidities were reported as being confused. The Glasgow components at the scene of trauma were largely at the highest functionality with almost all patients not requiring interventions (98.69%).

The FIM scores at discharge showed that the majority of patients regardless of comorbidity category were functionally independent with feeding and expression. However, with locomotion 45.86% of patients with comorbidities reported being dependent or needing help from a device, whereas, only 29.23% of patients without comorbidities needed help of some kind.

Chi-Squared tests were conducted at a 95% confidence level to determine if there was a significant difference between patients with comorbidities and those without in each variable category. Due to the large sample size of the data, all tests were significant and therefore p-values were not reported.

Table 8: (pg. 73)

Description:

Table 8 displays the range, mean, and standard deviation by comorbidity status for each numerical main variable.

Findings:

The maximum recorded patient age was 89. Patients older than 89 were recorded with -1 and have been set to missing in the *age* variable. However, these patients are reflected in the age range category (*agecat*) greater than 74. Ventilation days and the number of days in the ICU are supposed to be less than or equal to the length of hospital stay according to the data dictionary in Table 3. From the ranges, it is shown that this is accurate.

The mean patient age is 39 but the standard deviation is large (22.99). This data has a wide spread and appears to be positive or right skewed because of older patients present in the sample. The mean age of patients with comorbidities (52.32) is higher than the mean age of patients without (34.42). The mean length of stay is about 2.4 days shorter for patients without comorbidities. The number of ICU days and ventilation days shortens for patients without comorbidities by 1.09 and 0.78 days respectively. Functionality based variables appear approximately equally matched between patients with differing comorbidity statuses.

T-tests were conducted at a 95% confidence level to determine if there was a significant difference between patient comorbidity statuses for each variable. Due to the large sample size of the data, all tests were significant and p-values were therefore not reported.

Table 9: (pg. 74)

Description:

Table 9 shows the variable counts and percentages for categorical outcome main variables by patient age range. Percentages are given in three ways: the percentage of the overall variable data (%total), the percentage within the variable age category (%w/in), and the percentage within the variable category (%across).

Findings:

The majority of patients with a reported *disstatus* fell in the less than 55 age category (74.01%) with the next highest age category being greater than 74 (12.51%). There is a positive correlation between age category and death, the older the patient the higher the rate of death. At less than 55, 3.75% patients died, but at greater than 74, 9.19% died. The pattern with the FIM variables is that percents climb with age range for the lower functioning measures then taper out around measure 3 and then finally decrease in the over 74 age range at fully functioning.

Chi-Squared tests were conducted at a 95% confidence level to determine if there was a significant difference amongst patient age range categories for each variable category. Due to the large sample size of the data, all tests were significant and therefore p-values were not reported.

Table 10: (pg. 76)

Description:

Table 10 displays the means and standard deviations for numerical outcome main variables by patient age range.

Findings:

The noteworthy observation from Table 10 is that the patients with longest hospital length of stay, number of ICU days, and number of ventilation days are in the middle age categories, particularly in the 65 to 74 age range. The reasoning for this is unclear as it was expected that the length of time would climb with age. This is also unusual as the FIM total score is the lowest for the oldest age group (8.64), meaning that these patients would have needed the most assistance at discharge. Perhaps, hospital staff is less focused on prolonging care for those in the oldest age range due to the intensity of some medical procedures.

T-tests were conducted at a 95% confidence level to determine if there was a significant difference amongst patient age ranges for each variable. Due to the large sample size of the data, all tests were significant and therefore p-values were not reported.

Table 11: (pg. 77)

Description:

Table 11 provides the variable name for each comorbidity and comorbidity counts and percentages. Percentages are given by the percentage of each comorbidity type in the entire finalized dataset *Comorb.cleanedfinal* (%total) and the percentage of each comorbidity among the patients that are reported to have comorbidities (%comorbidities).

Findings:

Examining the percent totals for each pre-existing condition captures a picture of how well the sample data reflects the nation's population. For example, according to the CDC in the years 2002 to 2006 it is estimated that the American obesity rates ranged from a quarter to a third (www.cdc.gov/obesity). However, the sample population states that 1.15% of patients were obese. Clearly there is a discrepancy here. Either obese patients are less likely to incur a trauma or more likely, obesity was drastically underreported. Comorbidities that are related to previous surgeries or medical treatments, like history of cardiac surgery (3.05%), appear to give a closer representation of the general population. Chronic diseases were possibly not requested or recorded with the same vigor.

Table 12: (pg. 79)

Description:

Table 12 shows the comorbidity variable counts and percentages by patient age range. Percentages are given in three ways: the percentage of the total comorbidity data (%total), the percentage within the variable age category (%w/in), and the percentage within the comorbidity variable category (%across).

Findings:

It was previously computed that 926,450 patients are less than 55 years old, 96,028 are between 55 and 64 years old, 72,282 are between 65 and 74 years old, and 155,789 patients are over 74 years old. From this result and knowing the overall patient counts by age category with comorbidities (as listed in Table 12), it can be said that 18.47%, 41.37%, 49.67%, and 54.04% of patients in each age category respectively are reported as having pre-existing conditions.

Many of the results in Table 12 are as expected. There is a positive correlation between age and pre-existing conditions related to age like stroke with residual, coronary artery disease, congestive heart failure, chronic obstructive pulmonary disease, and chronic dementia. Other pre-existing conditions like chronic alcohol abuse and drug abuse are negatively correlated with age. These results are according to reason. The one comorbidity which is baffling is Alzheimer's disease. It decreases in percentage with age. This contradicts the results from chronic dementia.

Chi-Squared tests were conducted at a 95% confidence level to determine if there was a significant difference amongst patient age range categories for each comorbidity variable category. Due to the large sample size of the data, all tests were significant and therefore p-values were not reported.

Table 6

Main Variable Types, Missing Counts, and Missing Percentages

Variable	Variable Type	n missing	(%)
acslevel	predictor	45,933	(3.67%)
acspedlev	predictor	210,988	(16.87%)
acs_edrts	predictor	254,553	(20.36%)
acs_ps	predictor	221,372	(17.70%)
age	predictor	63,381	(5.07%)
disstatus	outcome	9,159	(0.73%)
edbasedef	predictor	1,058,184	(84.62%)
edgcseye	predictor	214,076	(17.12%)
edgcsmotor	predictor	201,546	(16.12%)
edgcstotal	predictor	217,708	(17.41%)
edgcsverb	predictor	216,314	(17.30%)
edrtrs	predictor	263,403	(21.06%)
edsysbp	predictor	117,466	(9.39%)
edtemp	predictor	476,411	(38.10%)
fimexpress	outcome	551,466	(44.10%)
fimfeed	outcome	549,023	(43.90%)
fimlocomt	outcome	550,004	(43.98%)
fimtotal	outcome	554,389	(44.33%)
gender	predictor	7,498	(0.60%)
ICUdays	outcome	436,195	(34.88%)
injurytype	predictor	19,843	(1.59%)
iss	predictor	63,487	(5.08%)
los	outcome	71,323	(5.70%)
payment	predictor	72,147	(5.77%)
race	predictor	89,835	(7.18%)
region	predictor	5,274	(0.42%)
sceneeye	predictor	717,477	(57.37%)
scenegcsaq	predictor	266,047	(21.27%)
scenemotor	predictor	717,903	(57.41%)
scenetotal	predictor	727,547	(58.18%)
scenever	predictor	726,180	(58.06%)
statelevel	predictor	125,600	(10.04%)
tempscale	predictor	337,503	(26.99%)
triss_prob	predictor	329,312	(26.33%)
ventdays	outcome	602,837	(48.21%)
yoadmit	predictor	0	(0.00%)

Table 7

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
acslevel	Overall	1,204,616	324,291	880,325
	%total	*100.00%*	*26.92%*	*73.08%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*26.92%*	*73.08%*
	Level I	435,857	96,668	339,189
	%total	*36.18%,*	*0.80%*	*28.16%*
	%w/in	*36.18%,*	*29.81%*	*38.53%*
	%across	*100.00%*	*2.22%*	*77.82%*
	Level II	234,825	63,707	171,118
	%total	*18.78%*	*5.29%*	*14.21%*
	%w/in	*18.78%*	*19.65%*	*19.44%*
	%across	*100.00%*	*27.13%*	*72.87%*
	Level III	30,910	9,442	21,468
	%total	*2.57%*	*0.78%*	*1.78%*
	%w/in	*2.57%*	*2.91%*	*2.44%*
	%across	*100.00%*	*30.55%*	*69.45%*
	Level IV	1,279	798	481
	%total	*0.11%*	*0.07%*	*0.04%*
	%w/in	*0.11%*	*0.25%*	*0.05%*
	%across	*100.00%*	*62.39%*	*37.61%*
	Not Applicable	501,745	153,676	348,069
	%total	*41.65%*	*12.76%*	*28.89%*
	%w/in	*41.65%*	*47.39%*	*39.54%*
	%across	*100.00%*	*30.63%*	*69.37%*
acspedlev	Overall	1,039,561	280,322	759,239
	%total	*100.00%*	*26.97%*	*73.03%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*26.97%*	*73.03%*
	Level I	131,153	21,507	109,646
	%total	*12.62%*	*2.06%*	*10.54%*
	%w/in	*12.62%*	*7.67%*	*14.44%*
	%across	*100.00%*	*16.40%*	*83.60%*
	Level II	34,800	9,774	25,026
	%total	*3.35%*	*2.95%*	*2.41%*
	%w/in	*3.35%*	*3.49%*	*3.30%*
	%across	*100.00%*	*28.09%*	*71.91%*
	Not Applicable	873,608	249,041	624,567
	%total	*84.04%*	*23.96%*	*60.08%*
	%w/in	*84.04%*	*88.84%*	*82.26%*
	%across	*100.00%*	*28.51%*	*71.49%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
disstatus	Overall	1,241,390	329,195	912,195
	%total	*100.00%*	*26.52%*	*73.48%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*26.52%*	*73.48%*
	Dead	58,760	19,697	39,063
	%total	*4.73%*	*1.59%*	*3.15%*
	%w/in	*4.73%*	*5.98%*	*4.28%*
	%across	*100.00%*	*33.52%*	*66.48%*
	Alive	1,182,630	309,498	873,132
	%total	*95.27%*	*24.93%*	*70.34%*
	%w/in	*95.27%*	*94.02%*	*95.27%*
	%across	*100.00%*	*24.93%*	*70.34%*
edgcseye	Overall	1,036,473	282,576	753,897
	%total	*100.00%*	*27.26%*	*72.74%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*27.26%*	*72.74%*
	1 = Does Not Open Eyes	89,554	24,102	65,452
	%total	*8.64%*	2.33%	*6.31%*
	%w/in	*8.64%*	*8.53%*	*8.68%*
	%across	*100.00%*	*26.91%*	*73.09%*
	2 = Opens Eyes to Pain	9,235	2,741	6,494
	%total	*0.89%*	0.26%	*0.63%*
	%w/in	*0.89%*	*0.97%*	*0.86%*
	%across	*100.00%*	*29.68%*	*70.32%*
	3 = Opens Eyes to Commands	24,355	7,627	16,728
	%total	*2.35%*	0.74%	*1.61%*
	%w/in	*2.35%*	*2.70%*	*2.22%*
	%across	*100.00%*	*31.32%*	*68.68%*
	4 = Spontaneous Eye Opening	913,329	248,106	665,223
	%total	*88.12%*	*23.94%*	*64.18%*
	%w/in	*88.12%*	*87.80%*	*88.24%*
	%across	*100.00%*	*27.17%*	*72.83%*
edgcsmotor	Overall	1,049,003	282,158	766,845
	%total	*100.00%*	*26.90%*	*73.10%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*26.90%*	*73.10%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
edgcsmotor (continued)	1 = None	75,495	19,617	55,878
	%total	*7.20%*	*1.87%*	*5.33%*
	%w/in	*7.20%*	*6.95%*	*7.29%*
	%across	*100.00%*	*25.98%*	*74.02%*
	2 = Extensor posturing in response to painful stimulation	3,156	929	2,227
	%total	*0.30%*	*0.09%*	*0.21%*
	%w/in	*0.30%*	*0.33%*	*0.29%*
	%across	*100.00%*	*29.44%*	*70.56%*
	3= Flexor posturing in response to painful stimulation	4,050	1,208	2,842
	%total	*0.39%*	*0.12%*	*0.27%*
	%w/in	*0.39%*	*0.43%*	*0.37%*
	%across	*100.00%*	*29.83%*	*70.17%*
	4 = General withdrawal in response to painful stimulation	13,853	4,180	9,673
	%total	*1.32%*	*0.40%*	*0.92%*
	%w/in	*1.32%*	*1.48%*	*1.26%*
	%across	*100.00%*	*30.17%*	*69.83%*
	5 = Localization of painful stimulation	28,335	9,063	19,272
	%total	*2.70%*	*0.86%*	*1.84%*
	%w/in	*2.70%*	*3.21%*	*2.51%*
	%across	*100.00%*	*31.99%*	*68.01%*
	6 = Obeys commands with appropriate motor response	924,114	247,161	676,953
	%total	*88.09%*	*23.56%*	*64.53%*
	%w/in	*88.09%*	*87.60%*	*88.28%*
	%across	*100.00%*	*26.75%*	*73.25%*
edgcsverb	Overall	1,034,235	282,038	752,197
	%total	*100.00%*	*27.27%*	*72.73%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*27.27%*	*72.73%*
	1 = None	93,396	25,868	67,528
	%total	*9.03%*	*2.50%*	*6.53%*
	%w/in	*9.03%*	*9.17%*	*8.98%*
	%across	*100.00%*	*27.70%*	*72.30%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
edgcsverb (continued)	2 = Incomprehensible words	10,486	3,363	7,123
	%total	*1.01%*	*0.33%*	*0.69%*
	%w/in	*1.01%*	*1.19%*	*0.95%*
	%across	*100.00%*	*32.07%*	*67.93%*
	3 = Inappropriate words	8,531	2,770	5,761
	%total	*0.82%*	*0.27%*	*0.56%*
	%w/in	*0.82%*	*0.98%*	*0.77%*
	%across	*100.00%*	*32.47%*	*67.53%*
	4 = Confused	76,143	26,043	50,100
	%total	*7.36%*	*2.52%*	*4.84%*
	%w/in	*7.36%*	*9.23%*	*6.66%*
	%across	*100.00%*	*34.20%*	*65.80%*
	5 = Oriented	845,679	223,994	621,685
	%total	*81.77%*	*21.66%*	*60.11%*
	%w/in	*81.77%*	*79.42%*	*82.65%*
	%across	*100.00%*	*26.49%*	*73.51%*
fimexpress	Overall	699,083	197,321	501,762
	%total	*100.00%*	*28.23%*	*71.77%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*28.23%*	*71.77%*
	1 = Dependent - Total Help Required	8,389	3,702	4,687
	%total	*1.20%*	*0.53%*	0.67%
	%w/in	*1.20%*	*1.88%*	*0.93%*
	%across	*100.00%*	*44.13%*	*55.87%*
	2 = Dependent - Partial Help Required	11,969	5,538	6,431
	%total	*1.71%*	0.79%	*0.92%*
	%w/in	*1.71%*	*2.81%*	*1.28%*
	%across	*100.00%*	*46.27%*	*53.73%*
	3 = Independent with Device	15,052	6,938	8,114
	%total	*2.15%*	*0.99%*	*1.16%*
	%w/in	*2.15%*	*3.52%*	*1.62%*
	%across	*100.00%*	*46.09%*	*53.91%*
	4 = Independent	562,991	162,524	400,467
	%total	*80.53%*	*23.25%*	*57.28%*
	%w/in	*80.53%*	*82.37%*	*79.81%*
	%across	*100.00%*	*28.87%*	*71.13%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
fimexpress (continued)	Not Applicable (<7 yrs or died)	100,682	18,619	82,063
	%total	*14.40%*	*2.66%*	*11.74%*
	%w/in	*14.40%*	*9.44%*	*16.35%*
	%across	*100.00%*	*18.49%*	*81.51%*
fimfeed	Overall	701,526	197,623	503,903
	%total	*100.00%*	*28.17%*	*71.83%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*28.17%*	*71.83%*
	1 = Dependent - Total Help Required	14,138	6,812	7,326
	%total	*2.02%*	*0.97%*	*1.04%*
	%w/in	*2.02%*	*3.45%*	*1.45%*
	%across	*100.00%*	*48.18%*	*51.82%*
	2 = Dependent - Partial Help Required	17,679	8,055	9,624
	%total	*2.52%*	*1.15%*	*1.37%*
	%w/in	*2.52%*	*4.08%*	*1.91%*
	%across	*100.00%*	*45.56%*	*54.44%*
	3 = Independent with Device	28,426	11,332	17,094
	%total	*4.05%*	*1.62%*	*2.44%*
	%w/in	*4.05%*	*5.73%*	*3.39%*
	%across	*100.00%*	*39.86%*	*60.14%*
	4 = Independent	540,325	152,771	387,554
	%total	*77.02%*	*21.78%*	*55.24%*
	%w/in	*77.02%*	*77.30%*	*76.91%*
	%across	*100.00%*	*28.27%*	*71.73%*
	Not Applicable (<7 yrs or died)	100,958	18,653	82,305
	%total	*14.39%*	*2.66%*	*11.73%*
	%w/in	*14.39%*	*9.44%*	*16.33%*
	%across	*100.00%*	*18.48%*	*81.52%*
fimlocomt	Overall	700,545	197,328	503,217
	%total	*100.00%*	*28.17%*	*71.83%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*28.17%*	*71.83%*
	1 = Dependent - Total Help Required	28,416	13,846	14,570
	%total	*4.06%*	*1.98%*	*2.08%*
	%w/in	*4.06%*	*7.02%*	*2.90%*
	%across	*100.00%*	*48.73%*	*51.28%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
fimlocomt (continued)	2 = Dependent - Partial Help Required	65,396	29,989	35,407
	%total	*9.34%*	*4.28%*	*5.05%*
	%w/in	*9.34%*	*15.20%*	*7.04%*
	%across	*100.00%*	*45.86%*	*54.14%*
	3 = Independent with Device	143,767	46,651	97,116
	%total	*20.52%*	*6.66%*	*13.86%*
	%w/in	*20.52%*	*23.64%*	*19.30%*
	%across	*100.00%*	*32.45%*	*67.55%*
	4 = Independent	362,402	88,251	274,151
	%total	*51.73%*	*12.60%*	*39.13%*
	%w/in	*51.73%*	*44.72%*	*54.48%*
	%across	*100.00%*	*24.35%*	*75.65%*
	Not Applicable (<7 yrs or died)	100,564	18,591	81,973
	%total	*14.36%*	*2.65%*	*11.70%*
	%w/in	*14.36%*	*9.42%*	*16.29%*
	%across	*100.00%*	*18.49%*	*81.51%*
gender	Overall	1,243,051	329,503	913,548
	%total	*100.00%*	*26.51%*	*73.49%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*26.51%*	*73.49%*
	Male	804,387	188,706	615,681
	%total	*64.71%*	*15.18%*	*49.53%*
	%w/in	*64.71%*	*57.27%*	*67.39%*
	%across	*100.00%*	*23.46%*	*76.54%*
	Female	438,664	140,797	297,867
	%total	*35.29%*	*11.33%*	*23.96%*
	%w/in	*35.29%*	*42.73%*	*32.61%*
	%across	*100.00%*	*32.10%*	*67.90%*
injurytype	Overall	1,230,706	327,418	903,288
	%total	*100.00%*	*26.60%*	*73.40%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*26.60%*	*73.40%*
	Blunt	1,077,336	294,617	782,719
	%total	*87.54%*	*23.94%*	*63.60%*
	%w/in	*87.54%*	*89.98%*	*86.65%*
	%across	*100.00%*	*27.35%*	*72.65%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
injurytype (continued)	Burn	26,688	6,380	20,308
	%total	*2.17%*	*0.52%*	*1.65%*
	%w/in	*2.17%*	*1.95%*	*2.25%*
	%across	*100.00%*	*23.91%*	*76.09%*
	Penetrating	126,682	26,421	100,261
	%total	*10.29%*	*2.15%*	*8.15%*
	%w/in	*10.29%*	*8.07%*	*11.10%*
	%across	*100.00%*	*20.86%*	*79.14%*
payment	Overall	1,178,402	311,912	866,490
	%total	*100.00%*	*26.47%*	*73.53%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*26.47%*	*73.53%*
	Automobile Insurance	49,523	11,779	37,744
	%total	*4.20%*	*1.00%*	*3.20%*
	%w/in	*4.20%*	*3.78%*	*4.36%*
	%across	*100.00%*	*23.78%*	*76.22%*
	Blue Cross/Blue Shield	49,826	10,232	39,594
	%total	*4.23%*	*0.87%*	*3.36%*
	%w/in	*4.23%*	*3.28%*	*4.57%*
	%across	*100.00%*	*20.54%*	*79.46%*
	CHAMPUS	4,136	754	3,382
	%total	*0.35%*	*0.06%*	*0.29%*
	%w/in	*0.35%*	*0.24%*	*0.39%*
	%across	*100.00%*	*18.23%*	*81.77%*
	Government/Military Insurance	8,381	1,547	6,834
	%total	*0.71%*	*0.13%*	*0.58%*
	%w/in	*0.71%*	*0.50%*	*0.79%*
	%across	*100.00%*	*18.46%*	*81.54%*
	Liability Insurance/Under Litigation	3,320	516	2,804
	%total	*0.28%*	*0.04%*	*0.24%*
	%w/in	*0.28%*	*0.17%*	*0.32%*
	%across	*100.00%*	*15.54%*	*84.46%*
	MCH and Crippled Children's	681	143	538
	%total	*0.06%*	*0.01%*	*0.05%*
	%w/in	*0.06%*	*0.05%*	*0.06%*
	%across	*100.00%*	*21.00%*	*79.00%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
payment (continued)	Managed Care Organization	99,704	22,159	77,545
	%total	*8.46%*	*1.88%*	*6.58%*
	%w/in	*8.46%*	*7.10%*	*8.95%*
	%across	*100.00%*	*22.22%*	*77.78%*
	Medicaid	106,435	29,954	76,481
	%total	*9.03%*	*2.54%*	*6.49%*
	%w/in	*9.03%*	*9.60%*	*8.83%*
	%across	*100.00%*	*28.14%*	*71.86%*
	Medicare	162,241	89,588	72,653
	%total	*13.77%*	*7.60%*	*6.17%*
	%w/in	*13.77%*	*28.72%*	*8.38%*
	%across	*100.00%*	*55.22%*	*44.78%*
	No Charge	11,844	2,154	9,690
	%total	*1.01%*	*0.18%*	*0.82%*
	%w/in	*1.01%*	*0.69%*	*1.12%*
	%across	*100.00%*	*18.19%*	*81.81%*
	No Fault Insurance	3,847	462	3,385
	%total	*0.33%*	*0.04%*	*0.29%*
	%w/in	*0.33%*	*0.15%*	*0.39%*
	%across	*100.00%*	*12.01%*	*87.99%*
	None	7,853	2,162	5,691
	%total	*0.67%*	*0.18%*	*0.48%*
	%w/in	*0.67%*	*0.69%*	*0.66%*
	%across	*100.00%*	*27.53%*	*72.47%*
	Not Done/Not Doc	96,866	18,475	78,391
	%total	*8.22%*	*1.57%*	*6.65%*
	%w/in	*8.22%*	*5.92%*	*9.05%*
	%across	*100.00%*	*19.07%*	*80.93%*
	Organ Donor Subsidy	21	2	19
	%total	*0.00%*	*0.00%*	*0.00%*
	%w/in	*0.00%*	*0.00%*	*0.00%*
	%across	*100.00%*	*9.52%*	*90.48%*
	Other	260,557	52,802	207,755
	%total	*22.11%*	*4.48%*	*17.63%*
	%w/in	*22.11%*	*16.93%*	*23.98%*
	%across	*100.00%*	*20.27%*	*79.73%*
	Other Commercial Indemnity Plan	105,253	28,647	76,606
	%total	*8.93%*	*2.43%*	*6.50%*
	%w/in	*8.93%*	*9.18%*	*8.84%*
	%across	*100.00%*	*27.22%*	*72.78%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
payment (continued)	Pending	998	112	886
	%total	*0.08%*	*0.01%*	*0.08%*
	%w/in	*0.08%*	*0.04%*	*0.10%*
	%across	*100.00%*	*11.22%*	*88.78%*
	Private Charity	734	323	411
	%total	*0.06%*	*0.03%*	*0.03%*
	%w/in	*0.06%*	*0.10%*	*0.05%*
	%across	*100.00%*	*44.01%*	*55.99%*
	Self-Pay	173,927	34,742	139,185
	%total	*14.76%*	*2.95%*	*11.81%*
	%w/in	*14.76%*	*11.14%*	*16.06%*
	%across	*100.00%*	*19.98%*	*80.02%*
	Worker's Compensation	32,255	5,359	26,896
	%total	*2.74%*	*0.45%*	*2.28%*
	%w/in	*2.74%*	*1.72%*	*3.10%*
	%across	*100.00%*	*16.61%*	*83.39%*
race	Overall	1,160,714	316,140	844,574
	%total	*100.00%*	*27.24%*	*72.76%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*27.24%*	*72.76%*
	Asian or Pacific Islander	14,570	2,717	11,853
	%total	*1.26%*	*0.23%*	*1.02%*
	%w/in	*1.26%*	*0.86%*	*1.40%*
	%across	*100.00%*	*18.65%*	*81.35%*
	Black	178,179	44,159	134,020
	%total	*15.35%*	*3.80%*	*11.55%*
	%w/in	*15.35%*	*13.97%*	*15.87%*
	%across	*100.00%*	*24.78%*	*75.22%*
	Hispanic	85,890	13,269	72,621
	%total	*7.40%*	*1.14%*	*6.26%*
	%w/in	*7.40%*	*4.20%*	*8.60%*
	%across	*100.00%*	*15.45%*	*84.55%*
	Native American or Alaskan Native	8,814	2,006	6,808
	%total	*0.76%*	*0.17%*	*0.59%*
	%w/in	*0.76%*	*0.63%*	*0.81%*
	%across	*100.00%*	*22.76%*	*77.24%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
race (continued)	White, not of Hispanic Origin	785,285	232,792	552,493
	%total	*67.66%*	*20.06%*	*47.60%*
	%w/in	*67.66%*	*73.64%*	*65.42%*
	%across	*100.00%*	*29.64%*	*70.36%*
	Other	87,976	21,197	66,779
	%total	*7.58%*	*1.83%*	*5.75%*
	%w/in	*7.58%*	*6.70%*	*7.91%*
	%across	*100.00%*	*24.09%*	*75.91%*
region	Overall	1,245,275	329,753	915,522
	%total	*100.00%*	*26.48%*	*73.52%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*26.48%*	*73.52%*
	Midwest	309,570	81,538	228,032
	%total	*24.86%*	*6.55%*	*18.31%*
	%w/in	*24.86%*	*24.73%*	*24.91%*
	%across	*100.00%*	*26.34%*	*73.66%*
	Northeast	253,569	9,6378	157,191
	%total	*20.36%*	*7.74%*	*12.62%*
	%w/in	*20.36%*	*29.23%*	*17.17%*
	%across	*100.00%*	*38.01%*	*61.99%*
	South	456,197	107,093	349,104
	%total	*36.63%*	*8.60%*	*28.03%*
	%w/in	*36.63%*	*32.48%*	*38.13%*
	%across	*100.00%*	*23.48%*	*76.52%*
	West	225,939	44,744	181,195
	%total	*18.14%*	*3.59%*	*14.55%*
	%w/in	*18.14%*	*13.57%*	*19.79%*
	%across	*100.00%*	*19.80%*	*80.20%*
sceneeye	Overall	533,072	151,023	382,049
	%total	*100.00%*	*28.33%*	*71.67%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*28.33%*	*71.67%*
	1 = None	44,741	10,821	33,920
	%total	*8.39%*	*2.03%*	*6.36%*
	%w/in	*8.39%*	*7.17%*	*8.88%*
	%across	*100.00%*	*24.19%*	*75.81%*
	2 = Pain	10,881	3,190	7,691
	%total	*2.04%*	*0.60%*	*1.44%*
	%w/in	*2.04%*	*2.11%*	*2.01%*
	%across	*100.00%*	*29.32%*	*70.68%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
sceneeye (continued)	3 = Voice	24,894	7,647	17,247
	%total	*4.67%*	*1.43%*	*3.24%*
	%w/in	*4.67%*	*5.06%*	*4.51%*
	%across	*100.00%*	*30.72%*	*69.28%*
	4 = Spontaneous	452,556	129,365	323,191
	%total	*84.90%*	*24.27%*	*60.63%*
	%w/in	*84.90%*	*85.66%*	*84.59%*
	%across	*100.00%*	*28.58%*	*71.41%*
scenegcsaq	Overall	984,502	279,794	704,708
	%total	*100.00%*	*28.42%*	*71.58%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*28.42%*	*71.58%*
	L = without interventions	971,564	276,314	695,250
	%total	*98.69%*	*28.07%*	*70.62%*
	%w/in	*98.69%*	*98.76%*	*98.66%*
	%across	*100.00%*	*28.44%*	*71.56%*
	S = Patient chemically sedated	2,955	914	2,041
	%total	*0.30%*	*0.09%*	*0.21%*
	%w/in	*0.30%*	*0.33%*	*0.29%*
	%across	*100.00%*	*30.93%*	*0.69%*
	T = Patient intubated	5,685	1,478	4,207
	%total	*0.58%*	*0.15%*	*0.43%*
	%w/in	*0.58%*	*0.53%*	*0.60%*
	%across	*100.00%*	*26.00%*	*74.00%*
	TP = Patient intubated and chemically paralyzed	4,298	1,088	3,210
	%total	*0.44%*	*0.11%*	*0.33%*
	%w/in	*0.44%*	*0.39%*	*0.46%*
	%across	*100.00%*	*25.31%*	*74.69%*
scenemotor	Overall	532,646	150,903	381,743
	%total	*100.00%*	*28.33%*	*71.67%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*28.33%*	*71.67%*
	1 = None	35,397	8,409	26,988
	%total	*6.65%*	*1.58%*	*5.07%*
	%w/in	*6.65%*	*5.57%*	*7.07%*
	%across	*100.00%*	*23.76%*	*76.24%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
scenemotor (continued)	2 = Extensor posturing in response to painful stimulation	2,952	874	2,078
	%total	*0.55%*	*0.16%*	*0.39%*
	%w/in	*0.55%*	*0.58%*	*0.54%*
	%across	*100.00%*	*29.61%*	*70.39%*
	3 = Flexor posturing in response to painful stimulation	4,908	1,427	3,481
	%total	*0.92%*	*0.27%*	*0.65%*
	%w/in	*0.92%*	*0.95%*	*0.91%*
	%across	*100.00%*	*29.07%*	*70.93%*
	4 = General withdrawal in response to painful stimulation	15,206	4,786	10,420
	%total	*2.85%*	*0.90%*	*1.96%*
	%w/in	*2.85%*	*3.17%*	*2.73%*
	%across	*100.00%*	*31.47%*	*68.53%*
	5 = Localization of painful stimulation	22,273	7,086	15,187
	%total	*4.18%*	*1.33%*	*2.85%*
	%w/in	*4.18%*	*4.70%*	*3.98%*
	%across	*100.00%*	*31.81%*	*68.19%*
	6 = Obeys commands with appropriate motor response	451,910	128,321	323,589
	%total	*84.84%*	*24.09%*	*60.75%*
	%w/in	*84.84%*	*85.04%*	*84.77%*
	%across	*100.00%*	*28.40%*	*71.60%*
scenever	Overall	524,369	148,877	375,492
	%total	*100.00%*	*28.39%*	*71.61%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*28.39%*	*71.61%*
	1 = None	38,389	9,601	28,788
	%total	*7.32%*	*1.83%*	*5.49%*
	%w/in	*7.32%*	*6.45%*	*7.67%*
	%across	*100.00%*	*25.01%*	*74.99%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
scenever (Continued)	2 =Incomprehensible Words	10,013	3,236	6,777
	%total	*1.91%*	*0.62%*	*1.29%*
	%w/in	*1.91%*	*2.17%*	*1.80%*
	%across	*100.00%*	*32.32%*	*67.68%*
	3 = Inappropriate Words	8,899	2,776	6,123
	%total	*1.70%*	*0.53%*	*1.17%*
	%w/in	*1.70%*	*1.86%*	*1.63%*
	%across	*100.00%*	*31.19%*	*6.88%*
	4 = Confused	68,606	21,732	46,874
	%total	*13.08%*	*4.14%*	*8.94%*
	%w/in	*13.08%*	*14.60%*	*12.48%*
	%across	*100.00%*	*31.68%*	*68.32%*
	5 = Oriented	398,462	111,532	286,930
	%total	*75.99%*	*21.27%*	54.72%
	%w/in	*75.99%*	*74.92%*	76.41%
	%across	*100.00%*	*27.99%*	72.01%
statelevel	Overall	1,124,949	302,547	822,402
	%total	*100.00%*	*26.89%*	*73.11%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*26.89%*	*73.11%*
	Level I	530,641	143,454	387,187
	%total	*47.17%*	*12.75%*	*34.42%*
	%w/in	*47.17%*	*47.42%*	*47.08%*
	%across	*100.00%*	*27.03%*	*72.97%*
	Level II	292,915	82,228	210,687
	%total	*26.04%*	*7.31%*	*18.73%*
	%w/in	*26.04%*	*27.18%*	*25.62%*
	%across	*100.00%*	*28.07%*	*71.93%*
	Level III	23,792	6,494	17,298
	%total	*2.11%*	*0.58%*	*1.54%*
	%w/in	*2.11%*	*2.15%*	*2.10%*
	%across	*100.00%*	*27.29%*	*72.71%*
	Level IV	2,104	1,047	1,057
	%total	*0.19%*	0.09%	0.09%
	%w/in	*0.19%*	0.35%	0.13%
	%across	*100.00%*	49.76%	50.24%
	Other	21,204	8,113	13,091
	%total	*1.88%*	*0.72%*	*1.16%*
	%w/in	*1.88%*	*2.68%*	*1.59%*
	%across	*100.00%*	*38.26%*	*61.74%*

*All tests found to be significant with $p < 0.0001$

Table 7 (Continued)

Categorical Variable Counts and Percentages based on Total, within Comorbidity Status, and across Variable Category

Variable	Variable Category	Overall *n (%, %, %)*	Comorbidities *n (%, %, %)*	No Comorbidities *n (%, %, %)*
statelevel	Not Applicable	254,293	61,211	193,082
	%total	*22.60%*	*5.44%*	*17.16%*
	%w/in	*22.60%*	*20.23%*	*23.48%*
	%across	*100.00%*	*24.07%*	*75.93%*
tempscale	Overall	913,046	252,230	660,816
	%total	*100.00%*	*27.63%*	*72.37%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*27.63%*	*72.37%*
	Celsius	643,124	149,250	493,874
	%total	*70.44%*	*16.35%*	*54.09%*
	%w/in	*70.44%*	*59.17%*	*74.74%*
	%across	*100.00%*	*23.21%*	*76.79%*
	Fahrenheit	269,922	102,980	166,942
	%total	*29.56%*	*11.28%*	*18.28%*
	%w/in	*29.56%*	*40.83%*	*25.26%*
	%across	*100.00%*	*38.15%*	*61.85%*
yoadmit	Overall	1,250,549	330,903	919,646
	%total	*100.00%*	*26.46%*	*73.54%*
	%w/in	*100.00%*	*100.00%*	*100.00%*
	%across	*100.00%*	*26.46%*	*73.54%*
	2002	200,649	43,695	156,954
	%total	*16.04%*	*3.49%*	*12.55%*
	%w/in	*16.04%*	*17.07%*	*13.20%*
	%across	*100.00%*	*21.78%*	*78.22%*
	2003	258,825	54,884	203,941
	%total	*20.70%*	*4.39%*	*16.31%*
	%w/in	*20.70%*	*22.18%*	*16.59%*
	%across	*100.00%*	*21.21%*	*78.79%*
	2004	250,756	57,756	193,000
	%total	*20.05%*	*4.62%*	*15.43%*
	%w/in	*20.05%*	*20.99%*	*17.45%*
	%across	*100.00%*	*23.03%*	*76.97%*
	2005	257,226	84,457	172,769
	%total	*20.57%*	*6.75%*	*13.82%*
	%w/in	*20.57%*	*18.79%*	*25.52%*
	%across	*100.00%*	*32.83%*	*67.17%*
	2006	283,093	90,111	192,982
	%total	*22.64%*	*7.21%*	*15.43%*
	%w/in	*22.64%*	*20.98%*	*27.23%*
	%across	*100.00%*	*31.83%*	*68.17%*

*All tests found to be significant with $p < 0.0001$

Table 8

Numerical Variable Ranges, Means, and Standard Deviations by Comorbidity Status

Variable	***Mean (Std Dev)***	***Min, Max***	**Comorbidities** ***Mean (Std Dev)***	**No Comorbidities** ***Mean (Std Dev)***
acs_edrts	7.35 (1.41)	(0.00, 7.48)	7.38 (1.30)	7.34 (1.45)
acs_ps	0.90 (0.26)	($1.34*10^{-6}$, 0.99)	0.93 (0.21)	0.89 (0.28)
age	39.14 (22.99)	(0, 89)	52.32 (22.45)	34.42 (21.29)
edbasedef	-1.64 (4.71)	(-80.00, 80.00)	-2.00 (5.07)	-1.48 (4.55)
edgcstotal	13.79 (3.27)	(3, 15)	13.76 (3.24)	13.80 (3.28)
edrtrs	7.21 (1.84)	(0, 8)	7.25 (1.74)	7.19 (1.88)
edsysbp	131.55 (35.83)	(0, 300)	137.40 (36.17)	129.40 (35.45)
edtemp	93.09 (19.26)	(0, 110)	94.18 (17.10)	92.66 (20.02)
fimtotal	9.51 (4.17)	(0, 12)	3.61 (0.01)	4.36 (0.01)
ICUdays	1.98 (6.08)	(0, 361)	2.75 (7.42)	1.66 (5.38)
iss	10.44 (9.88)	(0, 75)	11.54 (0.02)	10.03 (0.01)
los	5.32 (9.01)	(0, 362)	7.07 (10.79)	4.67 (8.17)
scenetotal	13.77 (3.03)	(3, 15)	13.83 (2.85)	13.74 (3.09)
triss_prob	0.94 (0.74)	(-5, 9)	0.97 (0.88)	0.93 (0.68)
ventdays	1.18 (5.21)	(0, 360)	1.71 (6.49)	0.93 (4.48)

*All tests found to be significant with $p < 0.0001$

Table 9

Categorical Outcome Variable Counts and Percentages based on Total, within Age Range, and across Variable Category

Outcome Variable	**Variable Category**	**< 55 years** ***n (%, %, %)***	**55 - 64 years** ***n (%, %, %)***	**65 - 74 years** ***n (%, %, %)***	**> 74 years** ***n (%, %, %)***
disstatus	Overall	918,740	95,414	71,926	155,310
	%total	*74.01%*	*7.69%*	*5.79%*	*12.51%*
	%w/in	*100.00%*	*100.00%*	*100.00%*	*100.00%*
	%across	*74.01%*	*7.69%*	*5.79%*	*12.51%*
	Alive	884,308	90,359	66,926	141,037
	%total	*71.24%*	*7.28%*	*5.39%*	*11.36%*
	%w/in	*96.25%*	*94.70%*	*93.05%*	*90.81%*
	%across	*74.77%*	*7.64%*	*5.66%*	*11.93%*
	Dead	34,432	5,055	5,000	14,273
	%total	*2.77%*	*0.41%*	*0.40%*	*1.15%*
	%w/in	*3.75%*	*5.30%*	*6.95%*	*9.19%*
	%across	*58.60%*	*8.60%*	*8.51%*	*24.29%*
fimexpress	Overall	528,258	53,078	38,808	78,939
	%total	*75.56%*	*7.59%*	*5.55%*	*11.29%*
	%w/in	*100.00%*	*100.00%*	*100.00%*	*100.00%*
	%across	*75.56%*	*7.59%*	*5.55%*	*11.29%*
	1	5,088	680	679	1,942
	%total	*0.73%*	*0.10%*	*0.10%*	*0.28%*
	%w/in	*0.96%*	*1.28%*	*1.75%*	*2.46%*
	%across	*60.65%*	*8.12%*	*8.09%*	*23.15%*
	2	6,264	952	1,027	3,726
	%total	*0.90%*	*0.14%*	*0.15%*	*0.53%*
	%w/in	*1.19%*	*1.79%*	*2.65%*	*4.72%*
	%across	*52.34%*	*7.95%*	*8.58%*	*31.13%*
	3	8,072	1,181	1,200	4,599
	%total	*1.15%*	*0.17%*	*0.17%*	*0.66%*
	%w/in	*1.53%*	*2.23%*	*3.09%*	*5.83%*
	%across	*53.63%*	*7.85%*	*7.97%*	*30.55%*
	4	428,081	45,523	31,740	57,647
	%total	*61.23%*	*6.51%*	*4.54%*	*8.25%*
	%w/in	*81.04%*	*85.77%*	*81.79%*	*73.03%*
	%across	*76.04%*	*80.86%*	*5.64%*	*10.24%*
	N/A	80,753	4,742	4,162	11,025
	%total	*11.55%*	*0.68%*	*0.60%*	*1.58%*
	%w/in	*15.29%*	*8.93%*	*10.72%*	*13.97%*
	%across	*80.21%*	*4.71%*	*41.34%*	*10.95%*

*All tests found to be significant with $p < 0.0001$

Table 9 (Continued)

Categorical Outcome Variable Counts and Percentages based on Total, within Age Range, and across Variable Category

Outcome Variable	**Variable Category**	**< 55 years** ***n (%, %, %)***	**55 - 64 years** ***n (%, %, %)***	**65 - 74 years** ***n (%, %, %)***	**> 74 years** ***n (%, %, %)***
fimfeed	Overall	530,168	53,229	38,904	79,225
	%total	*75.57%*	*7.59%*	*5.55%*	*11.29%*
	%w/in	*100.00%*	*100.00%*	*100.00%*	*100.00%*
	%across	*75.57%*	*7.59%*	*5.55%*	*11.29%*
	1	8,219	1,243	1,256	3,420
	%total	*1.17%*	*0.18%*	*0.18%*	*0.49%*
	%w/in	*1.55%*	*2.34%*	*3.23%*	*4.32%*
	%across	*58.13%*	*8.79%*	*8.88%*	*24.19%*
	2	8,911	1,435	1,598	5,735
	%total	*1.27%*	*0.20%*	*0.23%*	*0.82%*
	%w/in	*1.68%*	*2.70%*	*4.11%*	*7.24%*
	%across	*50.40%*	*8.12%*	*9.04%*	*32.44%*
	3	17,709	2,412	2,125	6,180
	%total	*2.52%*	*0.34%*	*0.30%*	*0.88%*
	%w/in	*3.34%*	*4.53%*	*5.46%*	*7.80%*
	%across	*62.30%*	*8.49%*	*7.48%*	*21.74%*
	4	414,354	43,385	29,754	52,832
	%total	*59.06%*	*6.18%*	*4.24%*	*7.53%*
	%w/in	*78.16%*	*81.51%*	*76.48%*	*66.69%*
	%across	*76.69%*	*8.03%*	*5.51%*	*9.78%*
	N/A	80,975	4,754	4,171	11,058
	%total	*11.54%*	*0.68%*	*0.59%*	*1.58%*
	%w/in	*15.27%*	*8.93%*	*10.72%*	*13.96%*
	%across	*80.21%*	*4.71%*	*4.13%*	*10.95%*
fimlocomt	Overall	529,478	53,122	38,834	79,111
	%total	*75.58%*	*7.58%*	*5.54%*	*11.29%*
	%w/in	*100.00%*	*100.00%*	*100.00%*	*100.00%*
	%across	*75.58%*	*7.58%*	*5.54%*	*11.29%*
	1	15,306	2,593	2,680	7,837
	%total	*2.18%*	*0.37%*	*0.38%*	*1.12%*
	%w/in	*2.89%*	*4.88%*	*6.90%*	*9.91%*
	%across	*53.86%*	*9.13%*	*9.43%*	*27.58%*
	2	29,863	6,283	7,147	22,103
	%total	*4.26%*	*0.90%*	*1.02%*	*3.16%*
	%w/in	*5.64%*	*11.83%*	*18.40%*	*27.94%*
	%across	*45.66%*	*9.61%*	*10.93%*	*33.80%*
	3	101,534	13,321	9,378	19,534
	%total	*14.49%*	*1.90%*	*1.34%*	*2.79%*
	%w/in	*19.18%*	*25.08%*	*24.15%*	*24.69%*
	%across	*70.62%*	*9.27%*	*6.52%*	*13.59%*

*All tests found to be significant with $p < 0.0001$

Table 9 (Continued)

Categorical Outcome Variable Counts and Percentages based on Total, within Age Range, and across Variable Category

Outcome Variable	**Variable Category**	**< 55 years** ***n (%, %, %)***	**55 - 64 years** ***n (%, %, %)***	**65 - 74 years** ***n (%, %, %)***	**> 74 years** ***n (%, %, %)***
fimlocomt	4	302,167	26,175	15,463	18,597
	%total	*43.13%*	*3.74%*	*2.21%*	*2.65%*
	%w/in	*57.07%*	*49.27%*	*39.82%*	*23.51%*
	%across	*83.38%*	*7.23%*	*4.27%*	*5.13%*
	N/A	80,608	4,750	4,166	11,040
	%total	*11.51%*	*0.69%*	*0.59%*	*1.58%*
	%w/in	*15.22%*	*8.94%*	*10.73%*	*13.69%*
	%across	*80.16%*	*4.72%*	*4.14%*	*10.98%*

*All tests found to be significant with p < 0.0001

Table 10

Numerical Outcome Variable Means and Standard Deviations by Age Range

Outcome Variable	**< 55 years** ***Mean (Std Dev)***	**55 to 64 years** ***Mean (Std Dev)***	**65 to 74 years** ***Mean (Std Dev)***	**> 74 years** ***Mean (Std Dev)***
fimtotal	9.59 (4.27)	10.03 (3.49)	9.53 (3.73)	8.64 (3.97)
ICUdays	1.87 (5.88)	2.56 (7.35)	2.71 (7.19)	1.92 (5.68)
los	4.86 (8.92)	6.63 (10.88)	6.84 (9.77)	6.41 (7.61)
ventdays	1.10 (4.95)	1.57 (6.37)	1.72 (6.61)	1.09 (5.06)

*All tests found to be significant with p < 0.0001

Table 11

Comorbidity Counts and Percentages based on Total and Total Comorbidity

Comorbidity	Variable	n (%total, %comorbidities)	
Acquired Coagulopathy	acqsum	1776	(0.14%, 0.54%)
Active Chemotherapy	chemosum	479	(0.04%, 0.14%)
Alzheimer’s Disease	alzhsum	7201	(0.58%, 2.18%)
Asthma	ashsum	33611	(2.69%, 10.16%)
Bilirubin > 2mg % (on Admission)	bilsum	380	(0.03%, 0.11%)
Chronic Alcohol Abuse	alchsum	42772	(3.42%, 12.93%)
Chronic Dementia	demsum	9938	(0.79%, 3.00%)
Chronic Demyelinating Disease	dmysum	203	(0.02%, 0.06%)
Chronic Drug Abuse	drugsum	24129	(1.93%, 7.29%)
Chronic Obstructive Pulmonary Disease	obplsum	18929	(1.51%, 5.72%)
Chronic Pulmonary Condition	pulcsum	6783	(0.54%, 2.05%)
Concurrent or Existence of Metastasis	metasum	3095	(0.25%, 0.94%)
Congestive Heart Failure	heartsum	19935	(1.59%, 6.02%)
Cor Pulmonale	corpsum	582	(0.05%, 0.18%)
Coronary Artery Disease	corartsum	36305	(2.90%, 10.97%)
Coumadin Therapy	coudsum	13339	(1.07%, 4.03%)
CVA/Hemiparesis (Stroke with Residual)	cvasum	11465	(0.92%, 3.46%)
Dialysis (Excludes Transplant Patients)	dialsum	2773	(0.22%, 0.84%)
Documented History of Cirrhosis	cirrsum	2519	(0.20%, 0.76%)
Documented Prior History of Pulmonary Disease with Ongoing Active Treatment	ppulsum	610	(0.05%, 0.18%)
Gastric or Esophageal Varices	gasvsum	1652	(0.13%, 0.50%)
Hemophilia	hemosum	375	(0.03%, 0.11%)
History of Cardiac Surgery	htycardsum	38087	(3.05%, 11.51%)
History of Psychiatric Disorders	psychsum	46606	(3.73%, 14.08%)
HIV/AIDS	hivsum	2047	(0.16%, 0.62%)
Hypertension	hpytsum	137818	(11.02%, 41.65%)
Inflammatory Bowel Disease	bowelsum	1035	(0.08%, 0.31%)
Insulin Dependent	insldepsum	17547	(1.40%, 5.30%)
Multiple Sclerosis	msclsum	937	(0.07%, 0.28%)
Non-Insulin Dependent	ninsldepsum	38187	(3.05%, 11.54%)
Obesity	obstsum	14362	(1.15%, 4.34%)
Organic Brain Syndrome	brainsum	1169	(0.09%, 0.35%)
Parkinsons Disease	parksum	3719	(0.30%, 1.12%)
Peptic Ulcer Disease	pepusum	2594	(0.21%, 0.78%)
Pre-existing Anemia	anemsum	14755	(1.18%, 4.46%)
Pregnancy	pregsum	5323	(0.43%, 1.61%)
Rheumatoid Arthritis	arthsum	5130	(0.41%, 1.55%)

Table 11 (Continued)

Comorbidity Counts and Percentages based on Total and Total Comorbidity

Comorbidity	Variable	n (%total, %comorbidities)
Routine Steroid Use	sterdsum	1070 (0.09%, 0.32%)
Seizures	seizsum	16641 (1.70%, 5.03%)
Serum Creatinine > 2 mg % (on Admission)	sercsum	232 (0.02%, 0.07%)
Spinal Cord Injury	spinalsum	6974 (0.56%, 2.11%)
Systemic Lupus Erythematous	lupussum	667 (0.05%, 0.20%)
Transplants	transpsum	934 (0.07%, 0.28%)
Undergoing Current Therapy	thpysum	2322 (0.19%, 0.70%)
Total Comorbidity	prescomorb	330903 (26.46%, 100.00%)

Table 12

Comorbidity Variable Counts and Percentages based on Total, within Age Range, and across Variable Category

Comorbidity	**< 55 years** ***n (%, %, %)***	**55 to 64 years** ***n (%, %, %)***	**65 to 74 years** ***n (%, %, %)***	**> 74 years** ***n (%, %, %)***
Acquired Coagulopathy	445	213	264	854
%total	*0.13%*	*0.06%*	*0.08%*	*0.26%*
%w/in	*0.26%*	*0.54%*	*0.74%*	*1.01%*
%across	*25.06%*	*11.99%*	*14.86%*	*48.09%*
Active Chemotherapy	91	96	116	176
%total	*0.03%*	*0.03%*	*0.04%*	*0.05%*
%w/in	*0.05%*	*0.24%*	*0.32%*	*0.21%*
%across	*19.00%*	*20.04%*	*24.22%*	*36.74%*
Alzheimer's Disease	6,195	620	162	224
%total	*1.87%*	*0.19%*	*0.05%*	*0.07%*
%w/in	*3.62%*	*1.56%*	*0.45%*	*0.27%*
%across	*86.03%*	*8.61%*	*2.25%*	*3.11%*
Asthma	26,638	2,400	1,703	2,870
%total	*8.05%*	*0.73%*	*0.51%*	*0.87%*
%w/in	*15.57%*	*6.04%*	*4.74%*	*3.41%*
%across	*79.25%*	*7.14%*	*5.07%*	*8.54%*
Bilirubin > 2mg % (on Admission)	258	67	29	26
%total	*0.08%*	*0.02%*	*0.01%*	*0.01%*
%w/in	*0.15%*	*0.17%*	*0.08%*	*0.03%*
%across	*67.89%*	*17.63%*	*7.63%*	*6.84%*
Chronic Alcohol Abuse	34,363	4,945	2,121	1,343
%total	*10.38%*	*1.49%*	*0.64%*	*0.41%*
%w/in	*20.08%*	*12.45%*	*5.91%*	*1.60%*
%across	*80.34%*	*11.56%*	*4.96%*	*3.14%*
Chronic Dementia	158	204	820	8,756
%total	*0.05%*	*0.06%*	*0.25%*	*2.65%*
%w/in	*0.09%*	*0.51%*	*2.28%*	*10.40%*
%across	*1.59%*	*2.05%*	*8.25%*	*88.11%*
Chronic Demyelinating Disease	88	27	28	60
%total	*0.03%*	*0.01%*	*0.01%*	*0.02%*
%w/in	*0.05%*	*0.07%*	*0.08%*	*0.07%*
%across	*43.35%*	*13.30%*	*13.79%*	*29.56%*
Chronic Drug Abuse	22,910	971	155	93
%total	*6.92%*	*0.29%*	*0.05%*	*0.03%*
%w/in	*13.39%*	*2.44%*	*0.43%*	*0.11%*
%across	*94.95%*	*4.02%*	*0.64%*	*0.39%*

*All tests found to be significant with $p < 0.0001$

Table 12 (Continued)

Comorbidity Variable Counts and Percentages based on Total, within Age Range, and across Variable Category

Comorbidity	**< 55 years** ***n (%, %, %)***	**55 to 64 years** ***n (%, %, %)***	**65 to 74 years** ***n (%, %, %)***	**> 74 years** ***n (%, %, %)***
Chronic Obstructive Pulmonary Disease	3,058	3,096	4,072	8,703
%total	*0.92%*	*0.94%*	*1.23%*	*2.63%*
%w/in	*1.79%*	*7.79%*	*11.34%*	*10.34%*
%across	*16.16%*	*16.36%*	*21.51%*	*45.98%*
Chronic Pulmonary Condition	2,464	912	1,100	2,307
%total	*0.74%*	*0.28%*	*0.33%*	*0.70%*
%w/in	*1.44%*	*2.30%*	*3.06%*	*2.74%*
%across	*36.33%*	*13.45%*	*16.22%*	*34.01%*
Concurrent or Existence of Metastasis	540	463	604	1,488
%total	*0.16%*	*0.14%*	*0.18%*	*0.45%*
%w/in	*0.32%*	*1.17%*	*1.68%*	*1.77%*
%across	*17.45%*	*14.96%*	*19.52%*	*48.08%*
Congestive Heart Failure	1,525	1,755	3,159	13,496
%total	*0.46%*	*0.53%*	*0.95%*	*4.08%*
%w/in	*0.89%*	*4.42%*	*8.80%*	*16.03%*
%across	*7.65%*	*8.80%*	*15.85%*	*67.70%*
Cor Pulmonale	78	51	95	358
%total	*0.02%*	*0.02%*	*0.03%*	*0.11%*
%w/in	*0.05%*	*0.13%*	*0.26%*	*0.43%*
%across	*13.40%*	*8.76%*	*16.32%*	*61.51%*
Coronary Artery Disease	4,080	5,125	7,338	19,762
%total	*1.23 %*	*1.55%*	*2.22%*	*5.97%*
%w/in	*2.38%*	*12.90%*	*20.44%*	*23.47%*
%across	*11.24%*	*14.12%*	*20.21%*	*54.43%*
Coumadin Therapy	1,290	1,386	2,527	8,136
%total	*0.40%*	*0.42%*	*0.76%*	*2.46%*
%w/in	*0.75%*	*3.49%*	*7.04%*	*9.66%*
%across	*9.67%*	*10.39%*	*18.94%*	*60.99%*
CVA/Hemiparesis (Stroke with Residual)	1,529	1,412	2,136	6,388
%total	*0.46%*	*0.43%*	*0.65%*	*1.93%*
%w/in	*0.89%*	*3.55%*	*5.95%*	*7.59%*
%across	*13.34%*	*12.32%*	*18.63%*	*55.72%*

*All tests found to be significant with $p < 0.0001$

Table 12 (Continued)

Comorbidity Variable Counts and Percentages based on Total, within Age Range, and across Variable Category

Comorbidity	< 55 years *n (%, %, %)*	55 to 64 years *n (%, %, %)*	65 to 74 years *n (%, %, %)*	> 74 years *n (%, %, %)*
Dialysis (Excludes Transplant Patients)	634	500	582	1,057
%total	*0.19%*	*0.15%*	*0.18%*	*0.32%*
%w/in	*0.37%*	*1.26%*	*1.62%*	*1.26%*
%across	*22.86%*	*18.03%*	*20.99%*	*38.12%*
Documented History of Cirrhosis	1,404	532	296	287
%total	*0.42%*	*0.16%*	*0.09%*	*0.09%*
%w/in	*0.82%*	*1.32%*	*0.82%*	*0.34%*
%across	*55.74%*	*21.12%*	*11.75%*	*11.39%*
Documented Prior History of Pulmonary Disease with Ongoing Active Treatment	325	72	67	146
%total	*0.10%*	*0.02%*	*0.02%*	*0.04%*
%w/in	*0.19%*	*0.18%*	*0.19%*	*0.17%*
%across	*53.28%*	*11.80%*	*10.98%*	*23.93%*
Gastric or Esophageal Varices	610	254	213	575
%total	*0.18%*	*0.08%*	*0.06%*	*0.17%*
%w/in	*0.36%*	*0.64%*	*0.59%*	*0.68%*
%across	*36.92%*	*15.38%*	*12.89%*	*34.81%*
Hemophilia	318	20	15	22
%total	*0.10%*	*0.01%*	*0.00%*	*0.01%*
%w/in	*0.19%*	*0.05%*	*0.04%*	*0.03%*
%across	*84.80%*	*5.33%*	*4.00%*	*5.87%*
History of Cardiac Surgery	17,520	4,219	5,044	11,304
%total	*5.29%*	*1.27%*	*1.52%*	*3.42%*
%w/in	*10.24%*	*10.62%*	*14.05%*	*13.43%*
%across	*46.00%*	*11.08%*	*13.24%*	*29.68%*
History of Psychiatric Disorders	30,147	5,287	3,517	7,655
%total	9.11%	1.60%	1.06%	2.31%
%w/in	17.62%	1.33%	9.80%	9.09%
%across	64.68%	11.34%	7.55%	16.42%
HIV/AIDS	1,837	158	35	17
%total	*0.56%*	*0.05%*	*0.01%*	*0.01%*
%w/in	*1.07%*	*0.40%*	*0.10%*	*0.02%*
%across	*89.74%*	*7.72%*	*1.71%*	*0.83%*

*All tests found to be significant with p < 0.0001

Table 12 (Continued)

Comorbidity Variable Counts and Percentages based on Total, within Age Range, and across Variable Category

Comorbidity	**< 55 years** ***n (%, %, %)***	**55 to 64 years** ***n (%, %, %)***	**65 to 74 years** ***n (%, %, %)***	**> 74 years** ***n (%, %, %)***
Hypertension	37,494	22,529	22,648	55,147
%total	*11.33%*	*6.81%*	*6.84%*	*16.67%*
%w/in	*21.91%*	*56.72%*	*63.08%*	*65.50%*
%across	*27.21%*	*16.35%*	*16.43%*	*40.01%*
Inflammatory Bowel Disease	420	149	133	333
%total	*0.13%*	*0.05%*	*0.04%*	*0.10%*
%w/in	*0.25%*	*0.38%*	*0.37%*	*0.40%*
%across	*40.58%*	*14.40%*	*12.85%*	*32.17%*
Insulin Dependent	6,035	3,512	3,359	4,641
%total	*1.82%*	*1.06%*	*1.02%*	*1.40%*
%w/in	*3.53%*	*8.84%*	*9.36%*	*5.51%*
%across	*34.39%*	*20.01%*	*19.14%*	*26.45%*
Multiple Sclerosis	506	236	109	86
%total	*0.15%*	*0.07%*	*0.03%*	*0.03%*
%w/in	*0.30%*	*0.59%*	*0.30%*	*0.10%*
%across	*54.00%*	*25.19%*	*11.63%*	*9.18%*
Non-Insulin Dependent	10,036	7,081	7,528	13,542
%total	*3.03%*	*2.14%*	*2.27%*	*4.09%*
%w/in	*5.87%*	*17.83%*	*20.97%*	*16.09%*
%across	*26.28%*	*18.54%*	*19.71%*	*35.46%*
Obesity	8,372	2,661	1,851	1,478
%total	*2.53%*	*0.80%*	*0.56%*	*0.45%*
%w/in	*4.89%*	*6.70%*	*5.16%*	*1.76%*
%across	*58.29%*	*18.53%*	*12.89%*	*10.29%*
Organic Brain Syndrome	140	55	110	864
%total	*0.04%*	*0.02%*	*0.03%*	*0.26%*
%w/in	*0.08%*	*0.14%*	*0.31%*	*1.03%*
%across	*11.98%*	*4.70%*	*9.41%*	*73.91%*
Parkinson's Disease	140	216	717	2,646
%total	0.04%	*0.07%*	*0.22%*	*0.80%*
%w/in	0.08%	*0.54%*	*2.00%*	*3.14%*
%across	3.76%	*5.81%*	*19.28%*	*71.15%*
Peptic Ulcer Disease	690	344	406	1,154
%total	*0.21%*	*0.10%*	*0.12%*	*0.35%*
%w/in	*0.40%*	*0.87%*	*1.13%*	*1.37%*
%across	*26.60%*	*9.57%*	*11.30%*	*32.11%*

*All tests found to be significant with $p < 0.0001$

Table 12 (Continued)

Comorbidity Variable Counts and Percentages based on Total, within Age Range, and across Variable Category

Comorbidity	**< 55 years** ***n (%, %, %)***	**55 to 64 years** ***n (%, %, %)***	**65 to 74 years** ***n (%, %, %)***	**> 74 years** ***n (%, %, %)***
Pre-existing Anemia	7,024	1,864	1,464	4,403
%total	*2.12%*	*0.56%*	*0.44%*	*1.33%*
%w/in	*4.11%*	*4.69%*	*4.08%*	*5.23%*
%across	*47.60%*	*12.63%*	*9.92%*	*29.84%*
Pregnancy	5,323	0	0	0
%total	*1.61%*	*0.00%*	*0.00%*	*0.00%*
%w/in	*3.11%*	*0.00%*	*0.00%*	*0.00%*
%across	*100.00%*	*0.00%*	*0.00%*	*0.00%*
Rheumatoid Arthritis	1,738	684	787	1,921
%total	*0.53%*	*0.21%*	*0.24%*	*0.58%*
%w/in	*1.02%*	*1.72%*	*2.19%*	*2.28%*
%across	*33.88%*	*13.33%*	*15.34%*	*37.45%*
Routine Steroid Use	268	156	229	417
%total	*0.08%*	*0.05%*	*0.07%*	*0.13%*
%w/in	*0.16%*	*0.39%*	*0.64%*	*0.50%*
%across	*25.05%*	*14.58%*	*21.40%*	*38.97%*
Seizures	10,854	2,063	1,374	2,350
%total	*3.28%*	*0.62%*	*0.42%*	*0.71%*
%w/in	*6.34%*	*5.19%*	*3.83%*	*2.79%*
%across	*65.22%*	*12.40%*	*8.26%*	*14.12%*
Serum Creatinine > 2 mg % (on Admission)	49	26	34	123
%total	*0.01%*	*0.01%*	*0.01%*	*0.04%*
%w/in	*0.03%*	*0.07%*	*0.09%*	*0.15%*
%across	*21.12%*	*11.21%*	*14.66%*	*53.02%*
Spinal Cord Injury	5,495	659	384	436
%total	*1.66%*	*0.20%*	*0.12%*	*0.13%*
%w/in	*3.21%*	*1.66%*	*1.07%*	*0.52%*
%across	*78.79%*	*9.45%*	*5.51%*	*6.25%*
Systemic Lupus Erythematous	352	134	88	93
%total	*0.11%*	*0.04%*	*0.03%*	*0.03%*
%w/in	*0.21%*	*0.34%*	*0.25%*	*0.11%*
%across	*52.77%*	*20.09%*	*13.19%*	*13.94%*

*All tests found to be significant with $p < 0.0001$

Table 12 (Continued)

Comorbidity Variable Counts and Percentages based on Total, within Age Range, and across Variable Category

Comorbidity	**< 55 years** ***n (%, %, %)***	**55 to 64 years** ***n (%, %, %)***	**65 to 74 years** ***n (%, %, %)***	**> 74 years** ***n (%, %, %)***
Transplants	486	201	145	102
%total	*0.15%*	*0.06%*	*0.04%*	*0.03%*
%w/in	*0.28%*	*0.51%*	*0.40%*	*0.12%*
%across	*52.03%*	*21.52%*	*15.52%*	*10.92%*
Undergoing Current Therapy	1,108	292	328	594
%total	*0.33%*	*0.09%*	*0.10%*	*0.18%*
%w/in	*0.65%*	*0.74%*	*0.91%*	*0.71%*
%across	*47.72%*	*12.58%*	*14.13%*	*25.58%*
Total Comorbidity	171,091	39,722	35,901	84,189
%total	*51.70%*	*12.00%*	*10.85%*	*25.44%*
%w/in	*100.00%*	*100.00%*	*100.00%*	*100.00%*
%across	*51.70%*	*12.00%*	*10.85%*	*25.44%*

*All tests found to be significant with $p < 0.0001$

II. Results of Data Investigation: Regression Analysis

After a picture of the data was captured by calculating descriptive statistics, inference was drawn by regression analysis. Tables 13 – 16 summarize the regression statistics obtained from the regression analysis stage of the methodology as described in Appendix A, "Analysis Part 6" (pg 131) and "Analysis Part 7" (pg 221). Regression analysis table descriptions and pertinent findings are as follows:

Table 13: (pg. 92)

Description:

Table 13 summarizes the main results of the thesis. It displays the results from the model building and regression analysis procedure for mortality estimation. It compares the unadjusted comorbidity model (model 1) to the comorbidity model with TRISS (model 2) to the comorbidity model with TRISS and age (model 3) for each pre-existing condition. Odds ratios and 95% confidence intervals are provided for all three models. NRI and IDI results are given for the inclusion of TRISS in model 1 creating model 2, and for the inclusion of age in model 2 creating model 3.

Findings:

The results for the unadjusted model show a significant effect of pre-existing condition on death outcome for all the comorbidities except:

1. Gastric Varices
2. HIV/AIDS
3. Inflammatory Bowel Disease
4. Multiple Sclerosis
5. Peptic Ulcer Disease

All significant comorbidities increased the risk of death expect:

1. Asthma
2. Chronic Drug Abuse
3. Chronic Alcohol Abuse
4. Hemophilia
5. History of psychiatric disorders
6. Pregnancy
7. Rheumatoid Arthritis
8. Seizures

These comorbidities, often chronic with ongoing treatment, appear to show an association with reduced mortality risk in the unadjusted model.

The inclusion of TRISS in model 2 was significant, or adds predictive ability, for all of the pre-existing conditions at a 95% significance level except:

Strict Criteria – not significant if either NRI or IDI fail:

1. Acquired Coagulopathy (NRI p-value: 0.3551)
2. Alzheimer's Disease (NRI p-value: 0.6157)
3. Chronic Demyelinating Disease (NRI p-value: 0.6096, IDI p-value: 0.1205)
4. Cor Pulmonale (IDI p-value: 0.1050)
5. CVA/Hemiparesis (NRI p-value: 0.1496)
6. Gastric Varices (NRI p-value: 0.3931, IDI p-value: 0.4501)
7. Inflammatory Bowel Disease (NRI no change, IDI p-value: 0.3609)
8. Multiple Sclerosis (NRI no change)
9. Obesity (NRI p-value: 0.1644, IDI p-value: 0.1846)
10. Organic Brain Syndrome (NRI p-value: 0.1049)
11. Parkinson's Disease (NRI p-value: 0.4884)
12. Peptic Ulcer Disease (IDI p-value: 0.2089)
13. Spinal Cord Injury (NRI p-value: 0.4612)
14. Systemic Lupus Erythematous (NRI: no change, IDI p-value: 0.1742)

Loose Criteria – not significant if NRI and IDI fail:

1. Chronic Demyelinating Disease (NRI p-value: 0.6096, IDI p-value: 0.1205)
2. Gastric Varices (NRI p-value: 0.3931, IDI p-value: 0.4501)
3. Inflammatory Bowel Disease (NRI no change, IDI p-value: 0.3609)
4. Systemic Lupus Erythematous (NRI: no change, IDI p-value: 0.1742)

Using model 2, the results show a significant effect of pre-existing condition on death outcome for all the comorbidities except:

1. Inflammatory Bowel Disease
2. Multiple Sclerosis

3. Peptic Ulcer Disease
4. Rheumatoid Arthritis
5. Seizures
6. Systemic Lupus Erythematous

All significant comorbidities increased the risk of death in model 2 except:

1. Asthma
2. Chronic Alcohol Abuse
3. Chronic Drug Abuse
4. Hemophilia
5. History of Psychiatric Disorders
6. Pregnancy

which show a level of protectiveness.

The inclusion of age in model 3 was significant, or adds predictive ability, for all of the pre-existing conditions at a 95% significance level except:

Strict Criteria – not significant if either NRI or IDI fail:

1. Active Chemotherapy (NRI p-value: 0.5949)
2. Alzheimer's Disease (NRI p-value: 0.6802)
3. Bilirubin > 2mg % (NRI p-value: 0.9023)
4. Chronic Dementia (NRI p-value: 0.6840)
5. Chronic Demyelinating Disease (NRI: no change, IDI: 0.9144)
6. Cor Pulmonale (NRI p-value: 0.4818)
7. Hemophilia (NRI: no change, IDI p-value: 0.3252)
8. HIV/AIDS (NRI p-value: 0.0730)
9. Insulin Dependent (NRI p-value: 0.1341, IDI p-value: 0.9482)
10. Non-Insulin Dependent (NRI p-value: 0.1101)
11. Obesity (NRI p-value: 0.1136)
12. Parkinson's Disease (IDI p-value: 0.8724)
13. Pre-existing Anemia (IDI p-value: 0.0967)
14. Routine Steroid Use (NRI p-value: 0.1968)
15. Serum Creatinine > 2 mg % (NRI p-value: 0.5020)
16. Systemic Lupus Erythematous (NRI: no change)
17. Transplants (NRI p-value: 0.5235)
18. Undergoing Current Therapy (NRI p-value: 0.2147, IDI p-value: 0.1931)

Loose Criteria – not significant if NRI and IDI fail:

1. Chronic Demyelinating Disease (NRI: no change, IDI: 0.9144)
2. Hemophilia (NRI: no change, IDI p-value: 0.3252)
3. Insulin Dependent (NRI p-value: 0.1341, IDI p-value: 0.9482)
4. Undergoing Current Therapy (NRI p-value: 0.2147, IDI p-value: 0.1931)

Using model 3, the results show a significant effect of pre-existing condition on death outcome for all the comorbidities except:

1. Alzheimer's Disease
2. Chronic Dementia
3. Chronic Demyelinating Disease
4. Cor Pulmonale
5. Gastric Varices
6. Hemophilia
7. Insulin Dependent
8. Non-Insulin Dependent
9. Obesity
10. Parkinson's disease
11. Systemic Lupus Erythematous
12. Undergoing Current Therapy

All significant comorbidities increased the risk of death in model 3 except:

1. Asthma
2. Chronic Alcohol Abuse
3. Chronic Drug Abuse
4. Chronic Pulmonary Condition
5. History of Psychiatric Disorders
6. Hypertension
7. Inflammatory Bowel Disease
8. Multiple Sclerosis
9. Peptic Ulcer Disease
10. Pregnancy
11. Rheumatoid Arthritis
12. Seizures

which show an association with reduced mortality risk.

The take-away point from Table 13 is that when considering the overall effect of comorbidity it is best to include TRISS and age due to significant NRI and IDI results for both inclusions. Also, the presence of comorbidities (Total Comorbidity, Model 3 - pg.95) significantly increases the risk of mortality after trauma by 9.8%. If selecting individual comorbidities for examination, pregnancy, chronic drug abuse, chronic alcohol abuse, asthma, and history of psychiatric disorders show the most beneficial odds ratios.

Acquired Coagulopathy, Bilirubin greater than 2mg % on admission, dialysis, history of cirrhosis, history of cardiac surgery, Organic Brain Syndrome, serum Creatinine greater than 2 mg % on admission, and spinal cord injury all increase the risk of mortality after trauma by over 200%.

Table 14: (pg. 96)

Description:

Table 14 describes the results of the regression analysis on model 4. Means and standard deviations by comorbidity status are provided for each comorbidity, as well as the adjusted comorbidity effect on length of hospital stay.

Findings:

Overall, due to the presence of comorbidities, a patient can expect a significantly longer length of hospital stay by just over 2 days than their healthy counterpart (2.39 days). All pre-existing conditions showed a significant adjusted comorbidity effect at a 95% confidence level except:

1. Active Chemotherapy
2. Asthma
3. Hemophilia
4. Multiple Sclerosis

With the exclusion of pregnancy, all significant comorbidities showed an increase in the amount of hospital time. Oddly, pregnancy significantly decreased the length of the hospital stay by 2.18 days. These women were probably transferred.

Table 15: (pg. 98)

Description:

Table 15 describes the results of the regression analysis on model 5. Means and standard deviations by comorbidity status are provided for each comorbidity, as well as the adjusted comorbidity effect on the number of days a patient spends in the ICU.

Findings:

Like length of stay, due to the presence of comorbidities, a patient can expect a significantly longer stay in the ICU by about a day than their healthy counterpart (1.07 days). All pre-existing conditions showed a significant adjusted comorbidity effect at a 95% confidence level except:

1. Active Chemotherapy
2. Bilirubin > 2mg %
3. Chronic Demyelinating Disease
4. Chronic Pulmonary Condition
5. Cor Pulmonale
6. Gastric Varices
7. Hemophilia
8. History of Psychiatric Disorders
9. Inflammatory Bowel Disease
10. Multiple Sclerosis
11. Organic Brain Syndrome
12. Peptic Ulcer Disease
13. Routine Steroid Use
14. Systemic Lupus Erythematous
15. Transplants

All significant comorbidities showed an increase in the amount of time in the ICU except for Alzheimer's disease, asthma, chronic dementia, pregnancy, and Rheumatoid Arthritis. These comorbidities produced a significant decrease in the number of ICU days.

Table 16: (pg. 100)

Description:

Table 16 describes the results of the regression analysis on model 6. Means and standard deviations by comorbidity status are provided for each comorbidity, as well as the adjusted comorbidity effect on the number of days a patient spends on breathing support.

Findings:

Like length of stay and the number of ICU days, due to the presence of comorbidities, a patient can expect a significantly longer length of time on ventilation support by about three-quarters of a day than their healthy counterpart (0.77 days). All pre-existing conditions showed a significant adjusted comorbidity effect at a 95% confidence level except:

1. Active Chemotherapy
2. Asthma
3. Chronic Demyelinating Disease
4. Chronic Pulmonary Condition
5. Cor Pulmonale
6. Gastric Varices
7. Hemophilia
8. History of Psychiatric Disorders
9. Inflammatory Bowel Disease
10. Multiple Sclerosis
11. Organic Brain Syndrome
12. Parkinson's Disease
13. Peptic Ulcer Disease
14. Systemic Lupus Erythematous
15. Transplants

All significant comorbidities showed an increase in the amount of ventilation support except for Alzheimer's disease, chronic dementia, pregnancy, and Rheumatoid Arthritis. These comorbidities produced a significant decrease in the number of ventilator days. It is unknown why these comorbidities produce this effect.

Table 13

Mortality Estimation Odds Ratios, Confidence Intervals, NRI, and IDI from Incremental Model Inclusions of Comorbidity, TRISS, and Age

Comorbidity	**Model 1 (Unadjusted)** ***OR (95% CI)***	**Model 2** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***	**Model 3** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***
Acquired Coagulopathy	3.745 (3.293, 4.259)	4.388 (3.845, 5.008) 0.0003 (0.3551) 0.0005 (0.0000)	2.890 (2.528, 3.304) 0.0028 (0.0000) 0.0003 (0.0000)
Active Chemotherapy	2.209 (1.635, 2.985)	2.225 (1.631, 3.036) -0.0011 (0.0000) 0.0000 (0.0000)	1.487 (1.091, 2.027) 0.0000 (0.5949) 0.0000 (0.0008)
Alzheimer’s Disease	1.883 (1.732, 2.047)	1.962 (1.801, 2.138) 0.0002 (0.6157) 0.0002 (0.0000)	1.026 (0.939, 1.121) 0.0000 (0.6802) -0.0000 (0.0055)
Asthma	0.445 (0.414, 0.479)	0.452 (0.419, 0.488) 0.0206 (0.0000) 0.0006 (0.0000)	0.488 (0.452, 0.526) 0.0045 (0.0000) 0.0005 (0.0000)
Bilirubin > 2mg % (on Admission)	3.218 (2.401, 4.312)	3.450 (2.562, 4.645) -0.0010 (0.0000) 0.0000 (0.0000)	3.049 (2.259, 4.116) 0.0000 (0.9023) 0.0000 (0.0000)
Chronic Alcohol Abuse	0.768 (0.730, 0.808)	0.776 (0.736, 0.818) -0.0021 (0.0000) 0.0001 (0.0000)	0.764 (0.725, 0.805) 0.0029 (0.0000) 0.0001 (0.0000)
Chronic Dementia	1.735 (1.612, 1.867)	1.824 (1.691, 1.968) 0.0011 (0.0481) 0.0002 (0.0000)	0.930 (0.861, 1.005) 0.0000 (0.6840) 0.0000 (0.0108)
Chronic Demyelinating Disease	1.854 (1.128, 3.047)	1.882 (1.122, 3.154) 0.0000 (0.6096) 0.0000 (0.1205)	1.424 (0.848, 2.391) . (.) -0.0000 (0.9144)
Chronic Drug Abuse	0.572 (0.529, 0.618)	0.572 (0.528, 0.620) 0.0124 (0.0000) 0.0002 (0.0000)	0.649 (0.599, 0.703) 0.0065 (0.0000) 0.0001 (0.0000)
Chronic Obstructive Pulmonary Disease	2.075 (1.974, 2.182)	2.162 (2.053, 2.276) 0.0031 (0.0000) 0.0005 (0.0000)	1.351 (1.281, 1.425) 0.0013 (0.0001) 0.0000 (0.0000)
Chronic Pulmonary Condition	1.249 (1.128, 1.384)	1.210 (1.090, 1.342) -0.0011 (0.0000) -0.0000 (0.0020)	0.861 (0.775, 0.956) 0.0010 (0.0000) 0.0000 (0.0000)
Concurrent or Existence of Metastasis	3.059 (2.755, 3.398)	3.137 (2.814, 3.497) 0.0009 (0.0250) 0.0004 (0.0000)	1.977 (1.775, 2.203) 0.0022 (0.0000) 0.0002 (0.0000)
Congestive Heart Failure	2.480 (2.369, 2.596)	2.616 (2.496, 2.743) 0.0058 (0.0000) 0.0011 (0.0000)	1.494 (1.423, 1.569) 0.0044 (0.0000) 0.0002 (0.0000)

Table 13 (Continued)

Mortality Estimation Odds Ratios, Confidence Intervals, NRI, and IDI from Incremental Model Inclusions of Comorbidity, TRISS, and Age

Comorbidity	**Model 1** **(Unadjusted)** ***OR (95% CI)***	**Model 2** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***	**Model 3** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***
Cor Pulmonale	2.558 (1.977, 3.310)	1.999 (1.535, 2.603) -0.0008 (0.0002) 0.0000 (0.1050)	1.197 (0.914, 1.567) -0.0001 (0.4818) -0.0000 (0.0654)
Coronary Artery Disease	2.145 (2.068, 2.224)	2.338 (2.253, 2.427) -0.0043 (0.0002) 0.0015 (0.0000)	1.403 (1.349, 1.459) 0.0064 (0.0000) 0.0003 (0.0000)
Coumadin Therapy	2.679 (2.539, 2.828)	2.783 (2.634, 2.940) 0.0023 (0.0023) 0.0013 (0.0000)	1.637 (1.548, 1.732) 0.0042 (0.0000) 0.0004 (0.0000)
CVA/Hemiparesis (Stroke with Residual)	1.933 (1.810, 2.064)	1.973 (1.845, 2.110) 0.0008 (0.1496) 0.0003 (0.0000)	1.191 (1.112, 1.274) 0.0006 (0.0031) 0.0000 (0.0004)
Dialysis (Excludes Transplant Patients)	2.961 (2.647, 3.312)	3.110 (2.772, 3.490) 0.0008 (0.0392) 0.0003 (0.0000)	2.060 (1.836, 2.311) 0.0022 (0.0000) 0.0002 (0.0000)
Documented History of Cirrhosis	4.595 (4.153, 5.084)	4.876 (4.394, 5.409) 0.0016 (0.0001) 0.0009 (0.0000)	3.941 (3.550, 4.376) 0.0096 (0.0000) 0.0008 (0.0000)
Documented Prior History of Pulmonary Disease with Ongoing Active Treatment	1.625 (1.200, 2.200)	1.686 (1.236, 2.299) -0.0010 (0.0000) 0.0000 (0.0001)	1.479 (1.084, 2.018) -0.0002 (0.0137) 0.0000 (0.0000)
Gastric or Esophageal Varices	1.183 (0.957, 1.461)	1.275 (1.029, 1.580) 0.0000 (0.3931) -0.0000 (0.4501)	0.900 (0.724, 1.114) 0.0002 (0.0043) 0.0000 (0.0003)
Hemophilia	0.442 (0.219, 0.891)	0.464 (0.230, 0.938) . (.) 0.0000 (0.0466)	0.535 (0.264, 1.082) . (.) 0.0000 (0.3252)
History of Cardiac Surgery	3.596 (3.490, 3.706)	3.561 (3.454, 3.671) -0.0104 (0.0000) 0.0070 (0.0000)	2.822 (2.736, 2.912) 0.0485 (0.0000) 0.0050 (0.0000)
History of Psychiatric Disorders	0.773 (0.736, 0.812)	0.804 (0.765, 0.845) -0.0012 (0.0024) 0.0001 (0.0000)	0.699 (0.665, 0.735) 0.0096 (0.0000) 0.0002 (0.0000)
HIV/AIDS	1.174 (0.971, 1.421)	1.231 (1.013, 1.495) -0.0017 (0.0000) 0.0000 (0.0007)	1.243 (1.024, 1.509) -0.0003 (0.0730) 0.0000 (0.0007)
Hypertension	1.228 (1.198, 1.259)	1.267 (1.235, 1.299) 0.0155 (0.0000) -0.0001 (0.0000)	0.763 (0.743, 0.784) 0.0100 (0.0000) 0.0011 (0.0000)

Table 13 (Continued)

Mortality Estimation Odds Ratios, Confidence Intervals, NRI, and IDI from Incremental Model Inclusions of Comorbidity, TRISS, and Age

Comorbidity	**Model 1** **(Unadjusted)** ***OR (95% CI)***	**Model 2** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***	**Model 3** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***
Inflammatory Bowel Disease	0.815 (0.594, 1.119)	0.890 (0.647, 1.223) . (.) -0.0000 (0.3609)	0.654 (0.475, 0.901) 0.0003 (0.0004) 0.0000 (0.0004)
Insulin Dependent	1.413 (1.329, 1.501)	1.468 (1.379, 1.563) -0.0033 (0.0000) 0.0001 (0.0000)	1.051 (0.987, 1.118) -0.0002 (0.1341) 0.0000 (0.9482)
Multiple Sclerosis	0.787 (0.562, 1.103)	0.711 (0.496, 1.017) . (.) 0.0000 (0.0090)	0.592 (0.413, 0.847) 0.0003 (0.0159) 0.0000 (0.0011)
Non-Insulin Dependent	1.407 (1.349, 1.467)	1.484 (1.422, 1.549) -0.0066 (0.0000) 0.0001 (0.0000)	0.977 (0.936, 1.021) 0.0003 (0.1101) 0.0000 (0.0000)
Obesity	1.132 (1.052, 1.218)	1.141 (1.058, 1.231) -0.0004 (0.1644) -0.0000 (0.1846)	0.974 (0.903, 1.052) 0.0002 (0.1136) 0.0000 (0.0000)
Organic Brain Syndrome	2.411 (2.001, 2.905)	3.832 (3.170, 4.634) -0.0005 (0.1049) 0.0001 (0.0000)	2.209 (1.823, 2.675) 0.0007 (0.0002) 0.0000 (0.0109)
Parkinson’s Disease	1.869 (1.664, 2.099)	2.003 (1.780, 2.255) -0.0003 (0.4884) 0.0001 (0.0000)	1.111 (0.987, 1.251) 0.0005 (0.0027) 0.0000 (0.8724)
Peptic Ulcer Disease	1.131 (0.952 , 1.344)	1.194 (1.000, 1.424) 0.0001 (0.0601) -0.0000 (0.2089)	0.780 (0.653, 0.931) 0.0004 (0.0016) 0.0000 (0.0000)
Pre-existing Anemia	1.749 (1.646, 1.858)	1.936 (1.820, 2.058) 0.0012 (0.0779) 0.0002 (0.0000)	1.319 (1.239, 1.404) 0.0031 (0.0000) -0.0000 (0.0967)
Pregnancy	0.355 (0.279, 0.425)	0.403 (0.39, 0.41) 0.0030 (0.0000) 0.0001 (0.0000)	0.458 (0.370, 0.568) 0.0042 (0.0000) 0.0000 (0.0000)
Rheumatoid Arthritis	0.844 (0.734, 0.970)	0.892 (0.773, 1.029) -0.0010 (0.0000) -0.0000 (0.0008)	0.619 (0.536, 0.715) 0.0007 (0.0000) 0.0001 (0.0000)
Routine Steroid Use	1.932 (1.562, 2.391)	2.110 (1.701, 2.617) -0.0008 (0.0022) 0.0000 (0.0000)	1.388 (1.119, 1.721) -0.0001 (0.1968) 0.0000 (0.0004)
Seizures	0.903 (0.837, 0.974)	0.951 (0.880, 1.027) -0.0006 (0.0001) 0.0000 (0.0000)	0.836 (0.774, 0.904) 0.0033 (0.0000) 0.0001 (0.0000)

Table 13 (Continued)

Mortality Estimation Odds Ratios, Confidence Intervals, NRI, and IDI from Incremental Model Inclusions of Comorbidity, TRISS, and Age

Comorbidity	**Model 1** **(Unadjusted)** ***OR (95% CI)***	**Model 2** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***	**Model 3** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***
Serum Creatinine > 2 mg % (on Admission)	3.760 (2.638, 5.359)	4.121 (2.866, 5.923) -0.0009 (0.0001) 0.0001 (0.0000)	2.598 (1.809, 3.73) 0.0001 (0.5020) 0.0001 (0.0000)
Spinal Cord Injury	2.970 (2.766, 3.189)	2.890 (2.689, 3.107) -0.0004 (0.4612) 0.0009 (0.0000)	2.951 (2.744, 3.174) 0.0146 (0.0000) 0.0010 (0.0000)
Systemic Lupus Erythematous	0.823 (0.556, 1.218)	0.830 (0.556, 1.237) . (.) 0.0000 (0.1742)	0.673 (0.451, 1.005) . (.) 0.0000 (0.0282)
Transplants	1.772 (1.399, 2.244)	1.877 (1.476, 2.388) -0.0013 (0.0000) 0.0000 (0.0026)	1.541 (1.213, 1.959) -0.0001 (0.5235) 0.0000 (0.0308)
Undergoing Current Therapy	1.323 (1.116, 1.568)	1.378 (1.158, 1.641) -0.0010 (0.0003) 0.0000 (0.0300)	1.105 (0.928, 1.316) -0.0000 (0.2147) 0.0000 (0.1931)
Total Comorbidity	1.427 (1.402, 1.452)	1.483 (1.456, 1.510) -0.1054 (0.0000) 0.0009 (0.0000)	1.098 (1.077, 1.120) -0.1054 (0.0000) 0.0009 (0.0000)

Table 14

Estimated Adjusted Comorbidity Effect on Length of Hospital Stay

Comorbidity	**Length of Stay No Comorbidities** *Mean (Std Dev)*	**Length of Stay Comorbidities** *Mean (Std Dev)*	**Adjusted Comorbidity Effect** *β2 (95% CI)*
Acquired Coagulopathy	5.31 (9.01)	7.99 (10.48)	3.33 (2.85, 3.82)
Active Chemotherapy	5.32 (9.02)	6.01 (6.81)	0.76 (-0.13, 1.65)
Alzheimer's Disease	5.31 (9.03)	6.32 (6.93)	1.19 (0.95, 1.43)
Asthma	5.32 (9.02)	5.30 (8.91)	-0.03 (-0.14, 0.08)
Bilirubin > 2mg % (on Admission)	5.32 (9.01)	7.52 (12.70)	1.84 (0.84, 2.85)
Chronic Alcohol Abuse	5.24 (8.88)	7.48 (12.01)	2.07 (1.97, 2.16)
Chronic Dementia	5.31 (9.03)	6.42 (6.37)	1.23 (1.03, 1.43)
Chronic Demyelinating Disease	5.32 (9.01)	7.33 (7.67)	1.94 (0.55, 3.34)
Chronic Drug Abuse	5.28 (8.97)	6.93 (10.97)	1.50 (1.37, 1.63)
Chronic Obstructive Pulmonary Disease	5.27 (8.98)	8.38 (10.78)	3.32 (3.17, 3.47)
Chronic Pulmonary Condition	5.31 (9.01)	6.82 (8.84)	1.42 (1.17, 1.66)
Concurrent or Existence of Metastasis	5.31 (9.01)	7.27 (10.08)	1.95 (1.59, 2.31)
Congestive Heart Failure	5.27 (8.98)	8.09 (10.36)	2.98 (2.83, 3.12)
Cor Pulmonale	5.31 (9.01)	9.35 (8.74)	3.70 (2.76, 4.64)
Coronary Artery Disease	5.25 (8.98)	7.47 (9.76)	2.30 (2.19, 2.40)
Coumadin Therapy	5.29 (9.00)	7.49 (9.71)	2.24 (2.08, 2.41)
CVA/Hemiparesis (Stroke with Residual)	5.29 (9.00)	7.83 (10.21)	2.46 (2.28, 2.65)
Dialysis (Excludes Transplant Patients)	5.31 (9.01)	8.66 (11.39)	3.18 (2.80, 3.57)
Documented History of Cirrhosis	5.31 (9.00)	9.93 (13.43)	4.52 (4.13, 4.92)
Documented Prior History of Pulmonary Disease with Ongoing Active Treatment	5.32 (9.02)	6.02 (7.83)	1.40 (0.57, 2.24)
Gastric or Esophageal Varices	5.31 (9.01)	6.68 (9.38)	1.36 (0.87, 1.86)
Hemophilia	5.32 (9.01)	5.86 (10.18)	0.83 (-0.23, 1.89)
History of Cardiac Surgery	5.11 (8.60)	11.62 (16.23)	6.49 (6.39, 6.59)

Table 14 (Continued)

Estimated Adjusted Comorbidity Effect on Length of Hospital Stay

Comorbidity	**Length of Stay No Comorbidities** *Mean (Std Dev)*	**Length of Stay Comorbidities** *Mean (Std Dev)*	**Adjusted Comorbidity Effect** *β2 (95% CI)*
History of Psychiatric Disorders	5.28 (9.00)	6.35 (9.40)	0.90 (0.81, 0.99)
HIV/AIDS	5.31 (9.01)	7.24 (11.55)	1.77 (1.32, 2.23)
Hypertension	5.13 (8.94)	6.82 (9.46)	1.69 (1.63, 1.74)
Inflammatory Bowel Disease	5.32 (9.02)	5.86 (7.23)	0.71 (0.10, 1.32)
Insulin Dependent	5.28 (8.97)	7.84 (11.24)	2.58 (2.43, 2.73)
Multiple Sclerosis	5.32 (9.02)	5.76 (6.48)	0.44 (-0.21, 1.10)
Non-Insulin Dependent	5.26 (8.97)	7.06 (10.13)	1.85 (1.75, 1.96)
Obesity	5.27 (8.96)	8.85 (12.28)	3.65 (3.48, 3.81)
Organic Brain Syndrome	5.31 (9.02)	7.01 (7.46)	2.24 (1.66, 2.82)
Parkinson’s Disease	5.31 (9.02)	6.84 (7.70)	1.68 (1.35, 2.01)
Peptic Ulcer Disease	5.31 (9.01)	7.02 (8.32)	1.85 (1.46, 2.24)
Pre-existing Anemia	5.27 (8.98)	9.24 (11.03)	4.05 (3.88, 4.21)
Pregnancy	5.33 (9.02)	3.14 (8.25)	-2.18 (-2.47, -1.90)
Rheumatoid Arthritis	5.31 (9.02)	5.91 (6.89)	1.07 (0.77, 1.38)
Routine Steroid Use	5.31 (9.01)	7.39 (9.25)	1.92 (1.34, 2.50)
Seizures	5.30 (9.00)	6.58 (10.18)	1.07 (0.91, 1.23)
Serum Creatinine > 2 mg % (on Admission)	5.32 (9.01)	10.01 (15.25)	5.61 (4.30, 6.91)
Spinal Cord Injury	5.27 (8.91)	13.86 (18.68)	8.63 (8.39, 8.87)
Systemic Lupus Erythematous	5.32 (9.01)	7.61 (16.86)	1.78 (1.01, 2.56)
Transplants	5.31 (9.01)	7.64 (12.10)	2.26 (1.60, 2.92)
Undergoing Current Therapy	5.31 (9.01)	6.06 (9.73)	0.62 (0.22, 1.02)
Total Comorbidity	4.67 (8.17)	7.07 (10.79)	2.39 (2.35, 2.42)

Table 15

Estimated Adjusted Comorbidity Effect on the Number of Patient ICU Days

Comorbidity	**ICU Days No Comorbidities** ***Mean (Std Dev)***	**ICU Days Comorbidities** ***Mean (Std Dev)***	**Adjusted Comorbidity Effect** ***β2 (95% CI)***
Acquired Coagulopathy	1.98 (6.07)	3.38 (7.33)	2.31 (1.97, 2.65)
Active Chemotherapy	1.98 (6.08)	1.93 (5.37)	0.05 (-0.58, 0.69)
Alzheimer's Disease	1.99 (6.08)	1.50 (4.82)	-0.44 (-0.62, -0.25)
Asthma	1.99 (6.09)	1.79 (5.77)	-0.21 (-0.29, -0.13)
Bilirubin > 2mg % (on Admission)	1.98 (6.07)	3.10 (8.64)	0.66 (-0.01, 1.33)
Chronic Alcohol Abuse	1.94 (6.00)	3.08 (7.63)	1.05 (0.98, 1.12)
Chronic Dementia	1.99 (6.09)	1.40 (3.99)	-0.41 (-0.56, -0.26)
Chronic Demyelinating Disease	1.98 (6.08)	2.18 (4.97)	0.32 (-0.67, 1.32)
Chronic Drug Abuse	1.97 (6.05)	2.64 (7.01)	0.57 (0.48, 0.66)
Chronic Obstructive Pulmonary Disease	1.96 (6.03)	3.45 (8.25)	1.48 (1.37, 1.59)
Chronic Pulmonary Condition	1.98 (6.07)	2.26 (6.49)	0.08 (-0.09, 0.26)
Concurrent or Existence of Metastasis	1.98 (6.07)	2.72 (6.89)	0.84 (0.58, 1.11)
Congestive Heart Failure	1.97 (6.05)	2.74 (7.33)	0.87 (0.76, 0.97)
Cor Pulmonale	1.98 (6.08)	2.87 (6.36)	0.29 (-0.37, 0.95)
Coronary Artery Disease	1.95 (6.02)	2.80 (7.41)	0.94 (0.87, 1.02)
Coumadin Therapy	1.97 (6.07)	2.76 (6.31)	0.88 (0.77, 1.00)
CVA/Hemiparesis (Stroke with Residual)	1.97 (6.07)	2.73 (6.85)	0.78 (0.65, 0.92)
Dialysis (Excludes Transplant Patients)	1.98 (6.07)	2.79 (7.53)	0.69 (0.41, 0.96)
Documented History of Cirrhosis	1.98 (6.07)	4.34 (8.12)	2.14 (1.86, 2.42)
Documented Prior History of Pulmonary Disease with Ongoing Active Treatment	1.98 (6.08)	2.44 (6.04)	0.88 (0.27, 1.48)
Gastric or Esophageal Varices	1.98 (6.07)	2.03 (7.56)	0.11 (-0.23, 0.45)
Hemophilia	1.98 (6.08)	2.00 (6.86)	0.24 (-0.50, 0.98)
History of Cardiac Surgery	1.83 (5.74)	5.76 (11.01)	3.89 (3.82, 3.96)

Table 15 (Continued)

Estimated Adjusted Comorbidity Effect on the Number of Patient ICU Days

Comorbidity	**ICU Days No Comorbidities** ***Mean (Std Dev)***	**ICU Days Comorbidities** ***Mean (Std Dev)***	**Adjusted Comorbidity Effect** ***β2 (95% CI)***
History of Psychiatric Disorders	1.98 (6.09)	2.07 (5.78)	0.01 (-0.06, 0.08)
HIV/AIDS	1.98 (6.07)	3.11 (7.82)	0.76 (0.42, 1.10)
Hypertension	1.92 (5.96)	2.43 (6.89)	0.47 (0.42, 0.51)
Inflammatory Bowel Disease	1.98 (6.08)	1.50 (4.49)	-0.27 (-0.71, 0.16)
Insulin Dependent	1.96 (6.04)	3.06 (8.11)	1.15 (1.04, 1.26)
Multiple Sclerosis	1.98 (6.08)	1.60 (4.76)	-0.44 (-0.93, 0.04)
Non-Insulin Dependent	1.97 (6.03)	2.48 (7.29)	0.53 (0.46, 0.61)
Obesity	1.96 (6.03)	3.50 (8.88)	1.53 (1.41, 1.65)
Organic Brain Syndrome	1.98 (6.08)	1.86 (5.09)	-0.07 (-0.59, 0.45)
Parkinson's Disease	1.98 (6.08)	2.12 (5.49)	0.27 (0.03, 0.51)
Peptic Ulcer Disease	1.98 (6.08)	2.09 (5.96)	0.19 (-0.10, 0.47)
Pre-existing Anemia	1.97 (6.05)	3.26 (7.68)	1.41 (1.29, 1.54)
Pregnancy	1.99 (6.08)	0.98 (5.52)	-1.12 (-1.33, -0.90)
Rheumatoid Arthritis	1.99 (6.08)	1.43 (4.37)	-0.34 (-0.57, -0.12)
Routine Steroid Use	1.98 (6.08)	2.25 (6.44)	0.18 (-0.20, 0.57)
Seizures	1.98 (6.08)	2.44 (6.28)	0.39 (0.27, 0.50)
Serum Creatinine > 2 mg % (on Admission)	1.98 (6.08)	3.97 (8.24)	2.68 (1.62, 3.74)
Spinal Cord Injury	1.95 (6.01)	6.69 (11.67)	4.68 (4.51, 4.84)
Systemic Lupus Erythematous	1.98 (6.07)	2.88 (12.43)	0.30 (-0.24, 0.85)
Transplants	1.98 (6.08)	2.33 (6.65)	0.37 (-0.09, 0.83)
Undergoing Current Therapy	1.98 (6.07)	2.43 (7.61)	0.38 (0.08, 0.67)
Total Comorbidity	1.66 (5.38)	2.75 (7.42)	1.07 (1.04, 1.10)

Table 16

Estimated Adjusted Comorbidity Effect on the Number of Patient Ventilation Days

Comorbidity	**Ventilation Days No Comorbidities** *Mean (Std Dev)*	**Ventilation Days Comorbidities** *Mean (Std Dev)*	**Adjusted Comorbidity Effect** *β2 (95% CI)*
Acquired Coagulopathy	1.18 (5.21)	2.24 (6.17)	1.56 (1.21, 1.91)
Active Chemotherapy	1.18 (5.21)	1.05 (4.61)	-0.06 (-0.65, 0.54)
Alzheimer's Disease	1.18 (5.21)	0.80 (4.20)	-0.40 (-0.56, -0.23)
Asthma	1.18 (5.21)	1.10 (5.28)	-0.06 (-0.13, 0.01)
Bilirubin > 2mg % (on Admission)	1.18 (5.21)	3.21 (9.66)	1.47 (0.60, 2.33)
Chronic Alcohol Abuse	1.15 (5.14)	1.88 (6.62)	0.73 (0.66, 0.80)
Chronic Dementia	1.18 (5.23)	0.64 (2.96)	-0.43 (-0.56, -0.29)
Chronic Demyelinating Disease	1.18 (5.21)	1.15 (4.90)	0.23 (-0.70, 1.17)
Chronic Drug Abuse	1.17 (5.19)	1.57 (5.76)	0.38 (0.29, 0.46)
Chronic Obstructive Pulmonary Disease	1.16 (5.14)	2.29 (8.02)	1.04 (0.94, 1.14)
Chronic Pulmonary Condition	1.18 (5.21)	1.39 (5.64)	0.13 (-0.04, 0.30)
Concurrent or Existence of Metastasis	1.18 (5.21)	1.76 (6.67)	0.67 (0.42, 0.93)
Congestive Heart Failure	1.17 (5.18)	1.64 (6.54)	0.49 (0.39, 0.59)
Cor Pulmonale	1.18 (5.21)	1.49 (5.39)	-0.43 (-1.04, 0.17)
Coronary Artery Disease	1.16 (5.15)	1.70 (6.46)	0.58 (0.51, 0.65)
Coumadin Therapy	1.17 (5.19)	1.72 (6.30)	0.61 (0.50, 0.72)
CVA/Hemiparesis (Stroke with Residual)	1.17 (5.20)	1.57 (5.87)	0.42 (0.30, 0.55)
Dialysis (Excludes Transplant Patients)	1.18 (5.20)	1.65 (6.69)	0.43 (0.17, 0.69)
Documented History of Cirrhosis	1.18 (5.20)	2.87 (7.60)	1.52 (1.26, 1.78)
Documented Prior History of Pulmonary Disease with Ongoing Active Treatment	1.18 (5.21)	1.59 (5.72)	0.88 (0.31, 1.46)
Gastric or Esophageal Varices	1.18 (5.21)	1.03 (4.92)	-0.08 (-0.38, 0.27)
Hemophilia	1.18 (5.21)	0.88 (4.48)	-0.14 (-0.84, 0.54)
History of Cardiac Surgery	1.05 (4.86)	4.05 (9.91)	2.98 (2.91, 3.04)

Table 16 (Continued)

Estimated Adjusted Comorbidity Effect on the Number of Patient Ventilation Days

Comorbidity	**Ventilation Days No Comorbidities** *Mean (Std Dev)*	**Ventilation Days Comorbidities** *Mean (Std Dev)*	**Adjusted Comorbidity Effect** *β2 (95% CI)*
History of Psychiatric Disorders	1.18 (5.22)	1.18 (4.95)	0.01 (-0.05, 0.07)
HIV/AIDS	1.18 (5.21)	1.93 (5.60)	0.73 (0.41, 1.05)
Hypertension	1.14 (5.07)	1.43 (6.08)	0.24 (0.20, 0.28)
Inflammatory Bowel Disease	1.18 (5.21)	1.00 (4.61)	-0.04 (-0.43, 0.35)
Insulin Dependent	1.16 (5.16)	1.91 (7.63)	0.81 (0.71, 0.92)
Multiple Sclerosis	1.18 (5.21)	0.92 (3.86)	-0.31 (-0.76, 0.13)
Non-Insulin Dependent	1.16 (5.15)	1.53 (6.58)	0.38 (0.30, 0.45)
Obesity	1.17 (5.15)	2.42 (8.11)	1.20 (1.09, 1.31)
Organic Brain Syndrome	1.18 (5.21)	0.59 (2.92)	-0.40 (-0.88, 0.09)
Parkinson’s Disease	1.18 (5.21)	1.18 (4.82)	0.02 (-0.20, 0.24)
Peptic Ulcer Disease	1.18 (5.21)	1.27 (5.14)	0.19 (-0.07, 0.44)
Pre-existing Anemia	1.17 (5.19)	1.86 (6.63)	0.73 (0.61, 0.84)
Pregnancy	1.18 (5.21)	0.46 (3.32)	-0.83 (-1.04, -0.62)
Rheumatoid Arthritis	1.18 (5.22)	0.75 (3.61)	-0.21 (-0.42, -0.01)
Routine Steroid Use	1.18 (5.21)	1.62 (6.85)	0.37 (0.03, 0.72)
Seizures	1.17 (5.20)	1.49 (5.69)	0.27 (0.17, 0.38)
Serum Creatinine > 2 mg % (on Admission)	1.18 (5.21)	3.49 (9.93)	2.61 (1.55, 3.67)
Spinal Cord Injury	1.15 (5.13)	4.97 (12.06)	3.60 (3.44, 3.77)
Systemic Lupus Erythematous	1.18 (5.21)	1.57 (8.92)	0.13 (-0.38, 0.63)
Transplants	1.18 (5.21)	1.38 (6.20)	0.07 (-0.35, 0.49)
Undergoing Current Therapy	1.18 (5.20)	1.88 (8.54)	0.64 (0.31, 0.98)
Total Comorbidity	0.93 (4.48)	1.71 (6.49)	0.77 (0.74, 0.80)

III. CAT Severity Score Testing Results

Regression analysis built the foundation for severity score development. This section depicts the CAT score content and its comparison to the original binary model. Tables 17 - 19, summarize the injury score testing information obtained from this stage of the methodology. The corresponding SAS code can be found in Appendix A, “Analysis Part 8” (pg 240). CAT severity score table descriptions and pertinent findings are as follows:

Table 17: (pg. 104)

Description:

Table 17 gives a selected sample of patients by their incident key to illustrate the CAT severity score they would receive with varying pre-existing conditions. Two CAT scores are presented for each patient – the unaltered CAT and the beneficial CAT. With the unaltered CAT score all odds ratios from the estimation results from model 3 were used in its creation. In the beneficial CAT score, all odds ratios that were less than one were set to one.

Findings:

Investigation into why some comorbidities are associated with reduced trauma mortality risk needs to be conducted before the unaltered or the beneficial score is chosen to represent the CAT scoring system. Clearly, there is a very real difference in these scores for some patients, as seen with *inc_key* 603193. A conclusion cannot be drawn until this is remedied.

<u>Table 18</u><u>:</u> (pg. 105)

<u>Description:</u>

Table 18 displays the results from the model building and regression analysis procedure for mortality estimation. It compares the unadjusted and full binary comorbidity models with *prescomorb* (model 1 & model 3) to the full score model with *comorscore* (model 7) and the full score model with *comorscore2* (model 8). Odds ratios and 95% confidence intervals are provided for all three models. NRI and IDI results are given for the inclusion of age to comorbidity and TRISS.

<u>Findings:</u>

Just like how the full binary model, as stated in Table 13, is best suited with both TRISS and age because of the significant p-values from NRI and IDI, the same can be concluded for both score models. All models show a significant increased risk in mortality for a patient with pre-existing conditions. However the score models show a higher risk than the unadjusted or binary models. Score model 1 or the unaltered CAT score model has the highest mortality risk of 27.3% greater than that of a healthy patient followed by the beneficial CAT score model (score model 2) with an 18.5% greater risk. The two score models are not drastically apart in their risk estimation but it is enough to beg for further investigation or analysis.

<u>Table 19</u><u>:</u> (pg. 105)

<u>Description:</u>

Table 19 describes the results of the regression analysis on models 9 - 14 which use *comorscore* and *comorscore2*. The adjusted comorbidity effect and 95% confidence interval for the length of hospital stay, number of ICU days, and number ventilation days are provided.

Findings:

It was found that each adjusted comorbidity effect is significant. Despite the high mortality risks given from the analysis of the score models in Table 18, the score adjusted comorbidity effects are lower than the binary model. All outcome lengths of time decrease from the binary model to the unaltered CAT score model to the beneficial CAT score model.

Table 17

CAT Severity Scoring System Comparison by Beneficial Odds Ratio Inclusion Status

Patient ID	Comorbidities Present	CAT Severity Score (beneficial ORs kept)	CAT Severity Score (No beneficial ORs)
603193	(1) Coronary Artery Disease (2) Pre-existing Anemia (3) Congestive Heart Failure (4) Hypertension (5) Insulin Dependent (6) Chronic Obstructive Pulmonary Disease (7) History of Psychiatric Disorders (8) Serum Creatinine > 2 mg % (on admission)	5.439	10.199
469045	(1) History of Cardiac Surgery (2) Spinal Cord Injury	8.327	8.327
468831	(1) History of Cardiac Surgery	2.822	2.822
603497	(1) Rheumatoid Arthritis (2) Hypertension (3) Insulin Dependent	0.496	1.051
520581	none	1.000	1.000

Table 18

Comparison of Mortality Estimation Odds Ratios, Confidence Intervals, NRI, and IDI Using Unadjusted Model, General Comorbidity Binary Model, and CAT Models

Unadjusted Model ***OR (95% CI)***	**Binary Model** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***	**Score Model (1)** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***	**Score Model (2)** ***OR (95% CI)*** ***NRI (p-value)*** ***IDI (p-value)***
1.018 (1.018, 1.019)	1.098 (1.077, 1.120) -0.1054 (0.0000) 0.0009 (0.0000)	1.273 (1.264, 1.283) 0.0454 (0.0000) 0.0054 (0.0000)	1.185 (1.177, 1.192) 0.0358 (0.0000) 0.0038 (0.0000)

Table 19

Comparison of Estimated Adjusted Comorbidity Effects on Outcome Variables Using the General Comorbidity Binary Model and CAT Models

Outcome Variable	**Binary Model: Adjusted Comorbidity Effect** ***β2 (95% CI)***	**Score Model (1): Adjusted Comorbidity Effect** ***β2 (95% CI)***	**Score Model (2): Adjusted Comorbidity Effect** ***β2 (95% CI)***
LOS	2.39 (2.35, 2.42)	1.38 (1.35, 1.40)	1.17 (1.14, 1.19)
ICUdays	1.07 (1.04, 1.10)	0.79 (0.77, 0.81)	0.61 (0.60, 0.63)
Ventdays	0.77 (0.74, 0.80)	0.60 (0.59, 0.62)	0.46 (0.44, 0.47)

UNEXPECTED PROBLEMS AND FINDINGS

Most of the problems that were encountered are rooted in intrinsic NTDB data issues. The descriptive statistic results that used *agecat* showed that several measures, like death or age-related pre-existing conditions, increase with age. However, NTDB records a patient older than 89 as -1 due to HIPPA laws, creating a potential bias for any analysis with *age* and comorbidity variables, as these patients were captured in *agecat* but set to missing in *age*. Regression analysis is therefore limited.

Another obstacle was the sample rates of certain pre-existing conditions, as reference in Table 11 findings. For example, the percentage of patients with recorded obesity in the revised and cleaned sample data was 1.15% and hypertension was 11.02%. As a precaution, to eliminate the possibility that removing certain facilities or manipulating comorbidity variables altered the data adversely, comorbidity percentages were captured from the originating NTDB dataset *rdscomorbid*. These results can be found in Table 20 below. Here, obesity is only reported at a mere 0.75% and hypertension 7.17%. As expected, all comorbidity percents are lower because the facilities with no recorded comorbidities were included, which lends credence to this project's methodology. It has to be an NTDB sample problem with underreporting.

Lastly, it was unexpected to find that any pre-existing condition would provide a mortality risk odds ratio less than one. Conditions like pregnancy and drug or alcohol abuse not only reduce the estimated mortality risk but shorten the length of time a patient spent in the hospital, ICU, and on ventilator support. This was unexpected and baffling.

Table 20

Original NTDB Comorbidity Counts and Percentages

Comorbidity	**n (% of Total)**
Acquired Coagulopathy	1,797 (0.09%)
Active Chemotherapy	484 (0.03%)
Alzheimer's Disease	7,227 (0.38%)
Asthma	33,697 (1.75%)
Bilirubin > 2mg % (on Admission)	381 (0.02%)
Chronic Alcohol Abuse	42,929 (2.23%)
Chronic Dementia	9,950 (0.52%)
Chronic Demyelinating Disease	207 (0.01%)

*Total source NTDB dataset observations: 1,926,245

Table 20 (Continued)

Original NTDB Comorbidity Counts and Percentages

Comorbidity	n	(% of Total)
Chronic Drug Abuse	24,252	(1.26%)
Chronic Obstructive Pulmonary Disease	20,117	(1.04%)
Chronic Pulmonary Condition	7,899	(0.41%)
Concurrent or Existence of Metastasis	3,100	(0.16%)
Congestive Heart Failure	20,147	(1.05%)
Cor Pulmonale	587	(0.03%)
Coronary Artery Disease	36,432	(1.89%)
Coumadin Therapy	13,360	(0.69%)
CVA/Hemiparesis (Stroke with Residual)	11,511	(0.60%)
Dialysis (Excludes Transplant Patients)	2,781	(0.14%)
Documented History of Cirrhosis	2,534	(0.13%)
Documented Prior History of Pulmonary Disease with Ongoing Active Treatment	614	(0.03%)
Gastric or Esophageal Varices	1,658	(0.09%)
Hemophilia	379	(0.02%)
History of Cardiac Surgery	38,241	(1.99%)
History of Psychiatric Disorders	46,685	(2.42%)
HIV/AIDS	2,047	(0.11%)
Hypertension	138,108	(7.17%)
Inflammatory Bowel Disease	1,035	(0.05%)
Insulin Dependent	17,726	(0.92%)
Multiple Sclerosis	1,023	(0.05%)
Non-Insulin Dependent	38,493	(2.00%)
Obesity	14,404	(0.75%)
Organic Brain Syndrome	1,170	(0.61%)
Parkinson’s Disease	3,725	(0.19%)
Peptic Ulcer Disease	2,598	(0.14%)
Pre-existing Anemia	14,929	(0.78%)
Pregnancy	5,360	(0.28%)
Rheumatoid Arthritis	5,144	(0.27%)
Routine Steroid Use	1,072	(0.06%)
Seizures	16,728	(0.87%)
Serum Creatinine > 2 mg % (on Admission)	232	(0.01%)
Spinal Cord Injury	7,260	(0.38%)
Systemic Lupus Erythematous	670	(0.03%)
Transplants	944	(0.05%)
Undergoing Current Therapy	2,363	(0.12%)

*Total source NTDB dataset observations

Chapter Five: Conclusions, Implications, and Recommendations

This chapter is a reflection on the research performed to develop the CAT injury scores and on injury scoring's role in medical care as a whole. Conclusions, potential implications, and recommendations for future research are provided.

CONCLUSIONS

There are two key conclusions that can be drawn from this research. One is that comorbidities, without a doubt, play a major role in the development of mortality risk. And second, the true extent to which these comorbidities affect mortality is still unclear. It was shown that the addition of comorbidity to mortality assessment models transfers additional predictive strength to models with age or age and TRISS. However, the variation in mortality risk was too vast between the binary and the CAT scoring models to draw an accurate conclusion as to the extent of the effect. More research needs to be conducted on the conditions producing beneficial odds ratios, and a unified approach to dealing with them in the creation of the score should be employed.

IMPLICATIONS FOR PRACTICE

Healthcare is a growing industry and with advancements in science and technology there is no sign of slowing. Novel techniques and effective, scientifically-backed guides for medical improvement are in demand. Injury severity scores have the potential to aid research, advance health treatment, and improve the patient quality of care. Chronic pre-existing conditions are on the rise and generation X may be the first generation to die younger than their parents (Belluck 2005). The CAT injury scores, with further development, could estimate mortality and hospital care outcomes for these groups. Also, many health care insurance practices and policy changes are anticipated and coming into effect. The CAT injury score could aid in determining benefit packages or governmental resource distribution.

RECOMMENDATIONS

Given further analysis, I would consider creating two more versions of the CAT scoring system and comparing them to the project's previous results. One version should be adapted from the NRI/IDI loose criteria and the other from the NRI/IDI strict criteria. My suggestions for which odds ratios from Table 13 to incorporate into the new versions are as follows:

- **Recreating OR choice for CAT Score using <u>Loose Criteria</u> (NRI *and* IDI non-significant) – change 7 ORs:**

Justified from non-significant NRI/IDI p-values from model 1:

5. Chronic Demyelinating Disease (OR: 1.854)
6. Gastric Varices (OR: 1.183)
7. Inflammatory Bowel Disease (OR: 0.815)
8. Systemic Lupus Erythematous (OR: 0.823)

Justified from non-significant NRI/IDI p-values from model 2:

5. Hemophilia (OR: 0.464)

6. Insulin Dependent (OR: 1.468)
7. Undergoing Current Therapy (OR: 1.378)

All other comorbidities maintain current ORs.

- **Recreating OR choice for Score using <u>Strict Criteria</u> (NRI *or* IDI non-significant) - change 25 ORS:**

Justified from non-significant NRI/IDI p-values from model 1:
15. Acquired Coagulopathy (OR: 3.745)
16. Alzheimer's Disease (OR: 1.883)
17. Chronic Demyelinating Disease (OR: 1.854)
18. Cor Pulmonale (OR: 2.558)
19. CVA/Hemiparesis (OR: 1.933)
20. Gastric Varices (OR: 1.183)
21. Inflammatory Bowel Disease (OR: 0.815)
22. Multiple Sclerosis (OR: 0.787)
23. Obesity (OR: 1.132)
24. Organic Brain Syndrome (OR: 2.411)
25. Parkinson's Disease (OR: 1.869)
26. Peptic Ulcer Disease (OR: 1.131)
27. Spinal Cord Injury (OR: 2.970)
28. Systemic Lupus Erythematous (OR: 0.823)

Justified from non-significant NRI/IDI p-values from model 2:
19. Active Chemotherapy (OR: 2.225)
20. Bilirubin > 2mg % (OR: 3.450)
21. Chronic Dementia (OR: 1.824)
8. Hemophilia (OR: 0.464)
22. HIV/AIDS (OR: 1.231)
23. Insulin Dependent (OR: 1.468)
24. Non-Insulin Dependent (OR: 1.484)
25. Pre-existing Anemia (OR: 1.936)
26. Routine Steroid Use (OR: 2.110)
27. Serum Creatinine > 2 mg % (OR: 4.121)
28. Transplants (OR: 1.877)
29. Undergoing Current Therapy (OR: 1.378)

All other comorbidities maintain current ORs.

My next recommendation would be to expand the models used to estimate mortality risk after trauma. I would suggest examining if the addition of other demographic variables would increase the model's predictive strength or if comorbidity interactions would affect mortality estimation. It might also be interesting to collect data that ranks a person's level of comorbidity severity, maybe on a five or 10 point scale, and apply it to the score development. Each pre-existing condition would therefore have five or ten comorbidity variables – one for each severity level. This would allow for patient differences in a pre-existing condition. I would also like to see other quality of life outcome indices tested like FIM scores.

In future CAT score development attempts and revisions I would advise for a a more stringent validation process. The CAT score needs to be applied to more datasets than just the NTDB data chosen for this project. Future editions of the NTDB dataset might have more critical variable dictionaries or stricter collection criteria. Data from sources other than NTDB should be used as well.

My last recommendation, and perhaps my most important one, is to always consider the person behind the number and the reality that although numbers and statistics may be good indicators or provide guidance, there are always exceptions to the norm. Patients are always in essence people, not a score or id. And even though one's identity may be removed and minimized when on paper, it is imperative to keep the human connection. Individuals should be handled just like the name implies - as individuals.

References

Baker, S. P., B. O'Neill, et al. (1974). The Injury Severity Score: A Method for Describing Patients with Multiple Injuries and Evaluating Emergency Care. The Journal of Trauma. **14:** 187-196.

Belluck, Pamela. “Children’s Life Expectancy being Cut Short by Obesity” (http://www.nytimes.com/2005/03/17/health/17obese.html) The New York Times 3/17/2005.

Boyd, C. R., M. A. Tolson, et al. (1987). "Evaluating Trauma Care: The TRISS Method." The Journal of Trauma **27**(4): 370-378.

Centers for Disease Control and Prevention. "http://www.cdc.gov/obesity/." Overweight and Obesity.

Champion, H. R., W. Sacco, et al. (1983). "Trauma severity scoring to predict mortality." World J. Surg. **7**(1): 4-11.

Champion, H. R., W. J. Sacco, et al. (1989). "A Revision of the Trauma Score." The Journal of Trauma **29**(5): 623-629.

Champion, H. R., W. S. Copes, et al. (1990). "A New Characterization of Injury Severity." The Journal of Trauma **30**(5): 539-546.

Charlson M, Pompei P, Ales K, MacKenzie C (1987). “A new method of classifying prognostic comorbidity in longitudinal studies: development and validation.” J Chron Dis **40**: 373–83.

Clark, D. E. and R. J. Winchell (2004). "Risk Adjustment for Injured Patients Using Administrative Data." The Journal of Trauma **57**(1): 130-140.

Fields, J. (2008). Commentary on Injury Severity Scoring and Outcomes Research. Trauma, McGraw-Hill Medical**:** 89-90.

Gartman, E. J., B. P. Casserly, et al. (2009). "Using serial severity scores to predict death in ICU patients: a validation study and review of the literature." Current Opinion in Critical Care **15**(6): 578-582

Kilgo, P. D., J. W. Meredith, et al. (2008). Injury Severity Scoring and Outcomes Research. Trauma, McGraw-Hill Medical**:** 83-88.

Linn, S. (1995). "The injury severity score—Importance and uses." Annals of epidemiology **5**(6): 440-446.

Moore, L., A. Lavoie, et al. (2008). "Using Information on Preexisting Conditions to Predict Mortality From Traumatic Injury." Annals of Emergency Medicine **52**(4): 356-364.

NTDB (2008). Resources for Evaluation of NTDB Data. National Trauma Data Bank, NTDB Research Data Set v 7.1: User Manual 9-11.

Osler, T. M. D. M., L. M. D. Glance, et al. (2008). "A Trauma Mortality Prediction Model Based on the Anatomic Injury Scale. [Article]." Annals of Surgery **247**(6): 1041-1048.

Rennie, C. P. and P. C. Brady (2007). "Advances in Injury Severity Scoring." Journal of emergency nursing: JEN : official publication of the Emergency Department Nurses Association **33**(2): 179-181.

Teasdale G., Murray G., Parker L. et al. (1979). “Adding up the Glasgow Coma Score.” Acta Neurochir Suppl (Wien) **28**:13.

APPENDIX A - SAS CODE:

```
*****************************************************************
*               C O M O R B I D I T Y   D A T A S E T            *
*                                                               *
*     CODE: Comorbidity_Project.sas                             *
*     PROGRAMMER: Rachel Rutkowski                              *
*     ADVISOR: Patrick Kilgo                                    *
*     CREATION DATE: May 24th, 2010                             *
*     LAST REVISION: March 6th, 2012                            *
*     CLASS: BIOS 595R (Practicum) / BIOS 599R (Thesis)         *
****************************************************************;

options nocenter nodate ls=72 ps=55 pageno=1;
*Libname Comorbid 'S:\bios\NTDB_RDS_2007_DBF\Comorbid';
Libname Comorbid 'U:\Practicum\NTDB_RDS_2007_DBF\Comorbid';
Libname temp 'U:\Practicum\NTDB_RDS_2007_DBF\temp';

*****************************************************************
*     ANALYSIS PART 1: Import and merge datasets                *
*                      RDS_CORMORBID.bdf and RDS_ED.bdf (To     *
*                      shorten run time may use reduced         *
*                      datasets short_CORMORBID.txt and         *
*                      short_ED.txt).                           *
*                      VARIABLES: INC_KEY = incidence key       *
*                        FAC_KEY = facility key(hospital)       *
*                        YOADMIT = year admitted               *
*                        PREXCOMOR = textual labels of          *
*                                    preexisting conditions     *
****************************************************************;

*"S:\bios\NTDB_RDS_2007_DBF\RDS_COMORBID.dbf";
*alternatively use short_COMORBID.txt for less obs;
Proc import OUT= rdscomorbid
                DATAFILE=
'U:\Practicum\NTDB_RDS_2007_DBF\Comorbid\RDS_COMORBID.dbf'
                DBMS=DBF REPLACE;
                GETDELETED=NO;
run;


*"S:\bios\NTDB_RDS_2007_DBF\RDS_ED.dbf";
*alternatively use short_ED.txt for less obs;
Proc import OUT= rdsed
                DATAFILE =
'U:\Practicum\NTDB_RDS_2007_DBF\Comorbid\RDS_ED.dbf'
                DBMS=DBF REPLACE;
                GETDELETED=NO;
run;
```

```
*"S:\bios\NTDB_RDS_2007_DBF\RDS_DEMO.dbf";
PROC IMPORT OUT= rdsdemo
            DATAFILE=
"U:\Practicum\NTDB_RDS_2007_DBF\Comorbid\RDS_DEMO.dbf"
            DBMS=DBF REPLACE;
           GETDELETED=NO;
RUN;

*"S:\bios\NTDB_RDS_2007_DBF\RDS_SCENE.dbf";
PROC IMPORT OUT= rdsscene
            DATAFILE=
"U:\Practicum\NTDB_RDS_2007_DBF\Comorbid\RDS_SCENE.dbf"
            DBMS=DBF REPLACE;
           GETDELETED=NO;
RUN;

*"S:\bios\NTDB_RDS_2007_DBF\RDS_DISCHARGE.dbf";
PROC IMPORT OUT= rdsdischarge
            DATAFILE=
"U:\Practicum\NTDB_RDS_2007_DBF\Comorbid\RDS_DISCHARGE.dbf"
            DBMS=DBF REPLACE;
           GETDELETED=NO;
RUN;

*"S:\bios\NTDB_RDS_2007_DBF\RDS_FACILITY.dbf";
PROC IMPORT OUT= rdsfacility
            DATAFILE=
"U:\Practicum\NTDB_RDS_2007_DBF\Comorbid\RDS_FACILITY.dbf"
            DBMS=DBF REPLACE;
           GETDELETED=NO;
RUN;

Proc sort data = rdscomorbid;
      By INC_KEY;
run;

Proc sort data = rdsed;
      By INC_KEY;
run;

***merged rdsed, rdscomorbid to have dataset to calculate percent
comorbidities;
Data temp.TempComorb;
      Merge rdsED rdsComorbid;
      By INC_KEY;
      Keep INC_KEY FAC_KEY YOADMIT PREXCOMOR;
run;
```

```
*****************************************************************
*    ANALYSIS PART 2: Merge imported datasets to create a final*
*         dataset without comorbidities in binary wide format. *
*          Will need to create a dummy variable to count the   *
*           number of comorbidities per INC_KEY.                *
*    VARIABLES: CombCount = comorbidity dummy variable(counter)*
*    INTEREST: PersonComb = total comorbidity count per patient*
*****************************************************************;


Data temp.comorb; *CombData.comorb;
     Set temp.TempComorb;
     By INC_KEY;
     ***creates a comorbidity counter;
     If PREXCOMOR=" " then CombCount=0;
     Else CombCount=1;
     Retain PersonComb 0;
     ***If there is no comorbidity for obs then comorbity total
     is zero;
     If CombCount=0 then PersonComb=0;
     ***If there is a comorbidity for obs then check if first
     obs for INC_KEY or repeated;
     If CombCount=1 then do;
          ***If first obs then comorbidity total is one;
          If first.INC_KEY then PersonComb = 1;
          ***If repeated obs then comorbidity total is added to
          previous count;
          else PersonComb = PersonComb + 1;
     end;
     If last.INC_KEY; *** Keeps only the last and final
     comorbidity count for each INC_KEY;
     if Personcomb > 0 then comorbidity=1;
          else comorbidity=0;
     Keep INC_KEY FAC_KEY YOADMIT PersonComb comorbidity;
run;

proc sort data= temp.comorb; *CombData.comorb;
     by fac_key yoadmit;
run;

Proc Means data= temp.comorb noprint; *CombData.comorb noprint;
     by fac_key yoadmit;
     Var comorbidity;
     output out=temp.meanout mean=mean n=n;
run;

data temp.meanout;
     set temp.meanout;
     if mean > 0;
run;
```

```
proc means sum;
      var _freq_;
run;

proc sort data=temp.meanout;
      by fac_key yoadmit;
run;

proc sort data=rdsED;
      by fac_key yoadmit;
run;

data temp.m1;
      merge rdsED (in=a) temp.meanout (in=b);
      by fac_key yoadmit;
      if b;
run;

proc sort data=temp.m1;
      by inc_key;
run;

proc sort data=rdsdemo;
      by inc_key;
run;

data temp.m2;
      merge temp.m1 (in=a) rdsdemo (in=b);
      by inc_key;
      if a;
run;

proc sort data=rdsdischarge;
      by inc_key;
run;

data temp.m3;
      merge temp.m2 (in=a) rdsdischarge (in=b);
      by inc_key;
      if a;
run;

proc sort data=rdsscene;
      by inc_key;
run;

data temp.m4;
      merge temp.m3 (in=a) rdsscene (in=b);
      by inc_key;
      if a;
run;
```

```
proc sort data = rdsfacility;
     by fac_key;
run;

proc sort data = temp.m4;
     by fac_key;
run;

data temp.m5;
     merge temp.m4 (in=a) rdsfacility (in=b);
     ***adding facility dataset for region and hospital type
     variables;
     by fac_key;
     if a;
run;

proc sort data = temp.m5;
     by inc_key;
run;

*****************************************************************
*    ANALYSIS PART 3:  Create binary comorbidity variables in  *
*    wide format and merge to create one final dataset. Store  *
*    final complete dataset in library called "Comorbid"       *
*    LOCATION:'S:\bios\NTDB_RDS_2007_DBF\Comorbid'              *
*****************************************************************;


*** create a dataset that just has pre-existing
conditions/comordities;
data temp.comortemp;
     set rdscomorbid;
     keep prexcomor;
run;

Proc sort data = temp.comortemp;
     by prexcomor;
run;

***Shows frequencies of just the comorbidities - would be great
resource for comorbidity chart;
Proc freq data = temp.comortemp;
run;

proc sort data=rdscomorbid;
     by inc_key;
run;

*proc print data = rdscomorbid (obs = 32);
*run;

***attempt1 was using proc iml;
```

```
***attempt2;
***create binary variables for comorbidities for logistic
regression purposes;
Data temp.bincomorb;
     set rdscomorbid;
     If prexcomor = "Acquired Coagulopathy" then acqcoag = 1;
          else acqcoag = 0;
     If prexcomor = "Active Chemotherapy" then activechemo = 1;
          else activechemo = 0;
     If prexcomor = "Alzheimers Disease" then alzheimers = 1;
          else alzheimers = 0;
     If prexcomor = "Asthma" then asthma = 1;
          else asthma = 0;
     If prexcomor = "Bilirubin > 2mg % (on Admission)" then
     bilirubin = 1;
          else bilirubin = 0;
     If prexcomor = "CVA/Hemiparesis (Stroke with Residual)"
     then cva = 1;
          else cva = 0;
     If prexcomor= "Chronic Alcohol Abuse" then alcoholabuse= 1;
          else alcoholabuse = 0;
     If prexcomor = "Chronic Dementia" then dementia = 1;
          else dementia = 0;
     If prexcomor = "Chronic Demyelinating Disease" then
     demyelinating = 1;
          else demyelinating = 0;
     If prexcomor = "Chronic Drug Abuse" then drugabuse = 1;
          else drugabuse = 0;
     If prexcomor = "Chronic Obstructive Pulmonary Disease" then
     obstpulmonary = 1;
          else obstpulmonary = 0;
     If prexcomor = "Chronic Pulmonary Condition" then
     pulmonarycond = 1;
          else pulmonarycond = 0;
     If prexcomor = "Concurrent or Existence of Metastasis" then
     metastasis = 1;
          else metastasis = 0;
     If prexcomor = "Congestive Heart Failure" then
     congheartfail = 1;
          else congheartfail = 0;
     If prexcomor = "Cor Pulmonale" then corpulmonale = 1;
          else corpulmonale = 0;
     If prexcomor= "Coronary Artery Disease" then coronartery=1;
          else coronartery = 0;
     If prexcomor = "Coumadin Therapy" then coumadinthpy = 1;
          else coumadinthpy = 0;
     If prexcomor = "Dialysis (Excludes Transplant Patients)"
     then dialysis = 1;
          else dialysis = 0;
     If prexcomor = "Documented History of Cirrhosis" then
     cirrhosis = 1;
```

```
        else cirrhosis = 0;
If prexcomor = "Documented Prior History of Pulmonary
Disease with Ongoing Active Treatment" then priorpulmonary
= 1;
        else priorpulmonary = 0;
If prexcomor = "Gastric or Esophageal Varices" then
gastricvarices = 1;
        else gastricvarices = 0;
If prexcomor = "HIV/AIDS" then hivaids = 1;
        else hivaids = 0;
If prexcomor = "Hemophilia" then hemophilia = 1;
        else hemophilia = 0;
If prexcomor = "History of Cardiac Surgery" then
histycardsurg = 1;
        else histycardsurg = 0;
If prexcomor = "History of Psychiatric Disorders" then
hstypsychiatric = 1;
        else hstypsychiatric = 0;
If prexcomor = "Hypertension " then hypertension = 1;
        else hypertension = 0;
If prexcomor = "Inflammatory Bowel Disease" then
inflambowel = 1;
        else inflambowel = 0;
If prexcomor = "Insulin Dependent" then insulindep = 1;
        else insulindep = 0;
If prexcomor = "Multiple Sclerosis" then multsclerosis = 1;
        else multsclerosis = 0;
If prexcomor= "Non-Insulin Dependent" then noninsulindep=1;
        else noninsulindep = 0;
If prexcomor = "Obesity" then obesity = 1;
        else obesity = 0;
If prexcomor= "Organic Brain Syndrome" then orgbrainsynd=1;
        else orgbrainsynd = 0;
If prexcomor = "Parkinsons Disease" then parkinsons = 1;
        else parkinsons = 0;
If prexcomor = "Peptic Ulcer Disease" then pepticulcer = 1;
        else pepticulcer = 0;
If prexcomor= "Pre-existing Anemia" then preexistanemia= 1;
        else preexistanemia = 0;
If prexcomor = "Pregnancy" then pregnancy = 1;
        else pregnancy = 0;
If prexcomor = "Rheumatoid Arthritis" then arthritis = 1;
        else arthritis = 0;
If prexcomor = "Routine Steroid Use" then steroiduse = 1;
        else steroiduse = 0;
If prexcomor = "Seizures" then seizures = 1;
        else seizures = 0;
If prexcomor = "Serum Creatinine > 2 mg % (on Admission)"
then serumcreatinine = 1;
        else serumcreatinine = 0;
If prexcomor= "Spinal Cord Injury" then spinalcordinjury=1;
        else spinalcordinjury = 0;
```

```
	If prexcomor = "Systemic Lupus Erythematous" then
	systemiclupus = 1;
		else systemiclupus = 0;
	If prexcomor = "Transplants" then transplants = 1;
		else transplants = 0;
	If prexcomor = "Undergoing Current Therapy" then
	undergoingthpy = 1;
		else undergoingthpy = 0;
run;

proc sort data = temp.bincomorb;
	by INC_KEY;
run;

Data temp.attempt2;
	set temp.bincomorb;
	by inc_key;
	Retain acqsum 0;
	if first.inc_key then acqsum=acqcoag;
		else acqsum=acqsum + acqcoag;
	Retain chemosum 0;
	if first.inc_key then chemosum=activechemo;
		else chemosum=chemosum + activechemo;
	Retain alzhsum 0;
	if first.inc_key then alzhsum=alzheimers;
		else alzhsum=alzhsum + alzheimers;
	Retain ashsum 0;
	if first.inc_key then ashsum=asthma;
		else ashsum=ashsum + asthma;
	Retain bilsum 0;
	if first.inc_key then bilsum=bilirubin;
		else bilsum=bilsum + bilirubin;
	Retain cvasum 0;
	if first.inc_key then cvasum=cva;
		else cvasum=cvasum + cva;
	Retain alchsum 0;
	if first.inc_key then alchsum=alcoholabuse;
		else alchsum=alchsum + alcoholabuse;
	Retain demsum 0;
	if first.inc_key then demsum=dementia;
		else demsum=demsum + dementia;
	Retain dmysum 0;
	if first.inc_key then dmysum=demyelinating;
		else dmysum=dmysum + demyelinating;
	Retain drugsum 0;
	if first.inc_key then drugsum=drugabuse;
		else drugsum=drugsum + drugabuse;
	Retain obplsum 0;
	if first.inc_key then obplsum=obstpulmonary;
		else obplsum=obplsum + obstpulmonary;
	Retain pulcsum 0;
	if first.inc_key then pulcsum=pulmonarycond;
```

```
      else pulcsum=pulcsum + pulmonarycond;
Retain metasum 0;
if first.inc_key then metasum=metastasis;
      else metasum=metasum + metastasis;
Retain heartsum 0;
if first.inc_key then heartsum=congheartfail;
      else heartsum=heartsum + congheartfail;
Retain corpsum 0;
if first.inc_key then corpsum=corpulmonale;
      else corpsum=corpsum + corpulmonale;
Retain corartsum 0;
if first.inc_key then corartsum=coronartery;
      else corartsum=corartsum + coronartery;
Retain coudsum 0;
if first.inc_key then coudsum=coumadinthpy;
      else coudsum=coudsum + coumadinthpy;
Retain dialsum 0;
if first.inc_key then dialsum=dialysis;
      else dialsum=dialsum + dialysis;
Retain cirrsum 0;
if first.inc_key then cirrsum=cirrhosis;
      else cirrsum=cirrsum + cirrhosis;
Retain ppulsum 0;
if first.inc_key then ppulsum=priorpulmonary;
      else ppulsum=ppulsum + priorpulmonary;
Retain gasvsum 0;
if first.inc_key then gasvsum=gastricvarices;
      else gasvsum=gasvsum + gastricvarices;
Retain hivsum 0;
if first.inc_key then hivsum=hivaids;
      else hivsum=hivsum + hivaids;
Retain hemosum 0;
if first.inc_key then hemosum=hemophilia;
      else hemosum=hemosum + hemophilia;
Retain htycardsum 0;
if first.inc_key then htycardsum=histycardsurg;
      else htycardsum=htycardsum + histycardsurg;
Retain psychsum 0;
if first.inc_key then psychsum=hstypsychiatric;
      else psychsum=psychsum + hstypsychiatric;
Retain hpytsum 0;
if first.inc_key then hpytsum=hypertension;
      else hpytsum=hpytsum + hypertension;
Retain bowelsum 0;
if first.inc_key then bowelsum=inflambowel;
      else bowelsum=bowelsum + inflambowel;
Retain insldepsum 0;
if first.inc_key then insldepsum=insulindep;
      else insldepsum=insldepsum + insulindep;
Retain msclsum 0;
if first.inc_key then msclsum=multsclerosis;
      else msclsum=msclsum + multsclerosis;
```

```
Retain ninsldepsum 0;
if first.inc_key then ninsldepsum=noninsulindep;
      else ninsldepsum=ninsldepsum + noninsulindep;
Retain obstsum 0;
if first.inc_key then obstsum=obesity;
      else obstsum=obstsum + obesity;
Retain brainsum 0;
if first.inc_key then brainsum=orgbrainsynd;
      else brainsum=brainsum + orgbrainsynd;
Retain parksum 0;
if first.inc_key then parksum=parkinsons;
      else parksum=parksum + parkinsons;
Retain pepusum 0;
if first.inc_key then pepusum=pepticulcer;
      else pepusum=pepusum + pepticulcer;
Retain anemsum 0;
if first.inc_key then anemsum=preexistanemia;
      else anemsum=anemsum + preexistanemia;
Retain pregsum 0;
if first.inc_key then pregsum=pregnancy;
      else pregsum=pregsum + pregnancy;
Retain arthsum 0;
if first.inc_key then arthsum=arthritis;
      else arthsum=arthsum + arthritis;
Retain sterdsum 0;
if first.inc_key then sterdsum=steroiduse;
      else sterdsum=sterdsum + steroiduse;
Retain seizsum 0;
if first.inc_key then seizsum=seizures;
      else seizsum=seizsum + seizures;
Retain sercsum 0;
if first.inc_key then sercsum=serumcreatinine;
      else sercsum=sercsum + serumcreatinine;
Retain spinalsum 0;
if first.inc_key then spinalsum=spinalcordinjury;
      else spinalsum=spinalsum + spinalcordinjury;
Retain lupussum 0;
if first.inc_key then lupussum=systemiclupus;
      else lupussum=lupussum + systemiclupus;
Retain transpsum 0;
if first.inc_key then transpsum=transplants;
      else transpsum=transpsum + transplants;
Retain thpysum 0;
if first.inc_key then thpysum=undergoingthpy;
      else thpysum=thpysum + undergoingthpy;

drop prexcomor acqcoag activechemo alzheimers asthma
bilirubin cva alcoholabuse dementia demyelinating drugabuse
obstpulmonary pulmonarycond metastasis congheartfail
corpulmonale coronartery coumadinthpy dialysis cirrhosis
priorpulmonary gastricvarices hivaids hemophilia
histycardsurg hstypsychiatric hypertension inflambowel
```

```
      insulindep multsclerosis noninsulindep obesity orgbrainsynd
      parkinsons pepticulcer preexistanemia pregnancy arthritis
      steroiduse seizures serumcreatinine spinalcordinjury
      systemiclupus transplants undergoingthpy;
run;

*proc freq data = temp.attempt2;
*run;

*proc print data=temp.attempt2 (obs=31);
*run;

Proc sort data = temp.attempt2;
      By INC_KEY;
run;

data temp.bincomorbid2;
      set temp.attempt2;
      by inc_key;
      if last.inc_key;
run;

*proc print data=temp.bincomorbid(obs=12);
*run;

data temp.m6;
      merge temp.m5 (in=a) temp.bincomorbid2 (in=b);
      by inc_key;
      if a;
run;

*proc print data= temp.m6 (obs=200);
      *var inc_key hpytsum;
*run;

data temp.all;
      set temp.m6;
      if acqsum="." then acqsum=0;
      if acqsum>1 then acqsum=1;
      if alchsum="." then alchsum=0;
      if alchsum>1 then alchsum=1;
      if alzhsum="." then alzhsum=0;
      if alzhsum>1 then alzhsum=1;
      if anemsum="." then anemsum=0;
      if anemsum>1 then anemsum=1;
      if arthsum="." then arthsum=0;
      if arthsum>1 then arthsum=1;
      if ashsum="." then ashsum=0;
      if ashsum>1 then ashsum=1;
      if bilsum="." then bilsum=0;
      if bilsum>1 then bilsum=1;
      if bowelsum="." then bowelsum=0;
```

```
if bowelsum>1 then bowelsum=1;
if brainsum="." then brainsum=0;
if brainsum>1 then brainsum=1;
if chemosum="." then chemosum=0;
if chemosum>1 then chemosum=1;
if cirrsum="." then cirrsum=0;
if cirrsum>1 then cirrsum=1;
if corartsum="." then corartsum=0;
if corartsum>1 then corartsum=1;
if corpsum="." then corpsum=0;
if corpsum>1 then corpsum=1;
if coudsum="." then coudsum=0;
if coudsum>1 then coudsum=1;
if cvasum="." then cvasum=0;
if cvasum>1 then cvasum=1;
if demsum="." then demsum=0;
if demsum>1 then demsum=1;
if dialsum="." then dialsum=0;
if dialsum>1 then dialsum=1;
if dmysum="." then dmysum=0;
if dmysum>1 then dmysum=1;
if drugsum="." then drugsum=0;
if drugsum>1 then drugsum=1;
if gasvsum="." then gasvsum=0;
if gasvsum>1 then gasvsum=1;
if heartsum="." then heartsum=0;
if heartsum>1 then heartsum=1;
if hemosum="." then hemosum=0;
if hemosum>1 then hemosum=1;
if hivsum="." then hivsum=0;
if hivsum>1 then hivsum=1;
if hpytsum="." then hpytsum=0;
if hpytsum>1 then hpytsum=1;
if htycardsum="." then htycardsum=0;
if htycardsum>1 then htycardsum=1;
if insldepsum="." then insldepsum=0;
if insldepsum>1 then insldepsum=1;
if lupussum="." then lupussum=0;
if lupussum>1 then lupussum=1;
if metasum="." then metasum=0;
if metasum>1 then metasum=1;
if msclsum="." then msclsum=0;
if msclsum>1 then msclsum=1;
if ninsldepsum="." then ninsldepsum=0;
if ninsldepsum>1 then ninsldepsum=1;
if obplsum="." then obplsum=0;
if obplsum>1 then obplsum=1;
if obstsum="." then obstsum=0;
if obstsum>1 then obstsum=1;
if parksum="." then parksum=0;
if parksum>1 then parksum=1;
if pepusum="." then pepusum=0;
```

```
      if pepusum>1 then pepusum=1;
      if ppulsum="." then ppulsum=0;
      if ppulsum>1 then ppulsum=1;
      if pregsum="." then pregsum=0;
      if pregsum>1 then pregsum=1;
      if psychsum="." then psychsum=0;
      if psychsum>1 then psychsum=1;
      if pulcsum="." then pulcsum=0;
      if pulcsum>1 then pulcsum=1;
      if seizsum="." then seizsum=0;
      if seizsum>1 then seizsum=1;
      if sercsum="." then sercsum=0;
      if sercsum>1 then sercsum=1;
      if spinalsum="." then spinalsum=0;
      if spinalsum>1 then spinalsum=1;
      if sterdsum="." then sterdsum=0;
      if sterdsum>1 then sterdsum=1;
      if thpysum="." then thpysum=0;
      if thpysum>1 then thpysum=1;
      if transpsum="." then transpsum=0;
      if transpsum>1 then transpsum=1;
      if disstatus='Dead' then dead=1;
           else dead=0;
run;


*proc print data=temp.all(Obs=200);
 *var inc_key hpytsum;
*run;


*CombData.final1;
***My final dataset gives 1250549 HOORAY!!;
data Comorbid.final;
      set temp.all;
      comorbsum = acqsum + alchsum + alzhsum + anemsum + arthsum
      + ashsum + bilsum + bowelsum + brainsum + chemosum +
      cirrsum + corartsum + corpsum + coudsum + cvasum + demsum +
      dialsum + dmysum + drugsum + gasvsum + heartsum + hemosum +
      hivsum + hpytsum + htycardsum + insldepsum + lupussum +
      metasum + msclsum + ninsldepsum + obplsum + obstsum +
      parksum + pepusum + ppulsum + pregsum + psychsum + pulcsum
      + seizsum + sercsum + spinalsum + sterdsum + thpysum +
      transpsum;
      if comorbsum > 0 then prescomorb=1;
           else prescomorb=0;
run;
```

```
****************************************************************
*    ANALYSIS PART 4:  Handling missing values and out of range*
*                       observations in the final dataset.      *
*                       Variables defined according to NTDB RDS *
*                       7.1 USER MANUAL                         *
***************************************************************;

*** checking current recorded ranges to see where out bounds need
to be coded;
*proc freq data=comorbid.final;
     *table yoadmit;
*run;

****Finish giving out of bounds settings;
data comorbid.cleanedfinal;
     set comorbid.final;
     if gender = "D" then gender= " ";
     if acspedlev = "N/A" then acspedlev = "Not Applicable";
     If edbasedef = -83.0 then edbasedef = .;
     If edbasedef = -81.0 then edbasedef = .;
     edgcstotal = edgcseye + edgcsverb + edgcsmotor;
     If edgcseye = . then edgcstotal = .;
     If edgcsmotor = . then edgcstotal = .;
     If edgcsverb = . then edgcstotal = .;
     If edgcstotal > 15 then edgcstotal = .;
     If edgcstotal < 3 then edgcstotal = .;
     If edrtrs = -5 then edrtrs=.;
     If edrtrs = 9 then edrtrs=.;
     If edsysbp > 300 then edsysbp = .;
     If edsysbp < 0 then edsysbp = .;
     If tempscale = "C" then edtemp = (edtemp*1.8) + 32;
     If edtemp > 110 then edtemp = .;
     If edtemp < 0 then edtemp =.;
     If Fimexpress = 9 then Fimexpress = .;
     If Fimexpress = 8 then Fimexpress = 0;
     If Fimfeed = 9 then Fimfeed = .;
     If Fimfeed = 8 then Fimfeed = 0;
     If Fimlocomt = 9 then Fimlocomt = .;
     If Fimlocomt = 8 then Fimlocomt = 0;
     Fimtotal = Fimexpress + Fimfeed + Fimlocomt;
     If Fimexpress = . then Fimtotal = .;
     If Fimfeed = . then Fimfeed = .;
     If Fimlocomt = . then Fimlocomt = .;
     If Los > 364 then Los = .;
     If Los < 0 then Los = .;
     If ICUDays < 0 then ICUDays = .;
     If ICUDays > Los then ICUDays = .;
     If Ventdays < 0 then Ventdays = .;
     If Ventdays > Los then Ventdays = .;
     If iss > 75 then iss = .;
     If iss < 0 then iss = .;
     If scenemotor = 9 then scenemotor = .;
```

```
      If scenever = 9 then scenever = .;
      If scenegcsaq = "T" then scenever = .;
      If scenegcsaq = "TP" then scenever = .;
      scenetotal = scenemotor + scenever + sceneeye;
      If sceneeye = . then scenetotal = .;
      If scenemotor = . then scenetotal = .;
      If scenever = . then scenetotal = .;
      ***creating age categories for comorbidity table;
      if age<55 then agecat=1;
      if (age>54) & (age<65) then agecat=2;
      if (age>64) & (age<75) then agecat=3;
      if age>74 then agecat=4;
      if age = -1 then agecat=4;
      if age < 0 then age = .;
run;

proc sort data= comorbid.cleanedfinal;
      by agecat pregsum;
run;

proc print data = comorbid.cleanedfinal;
      where (agecat=4) & (pregsum=1);
      var pregsum agecat comorbsum;
run;

data comorbid.cleanedfinal;
      set comorbid.cleanedfinal;
      if (agecat=2) & (pregsum=1) then pregsum=0;
      if (agecat=3) & (pregsum=1) then pregsum=0;
      if (agecat=4) & (pregsum=1) then pregsum=0;
run;

proc sort data= comorbid.cleanedfinal;
      by inc_key;
run;


*****************************************************************
*     ANALYSIS PART 5:   Making demographic and comorbidity     *
*     tables (Excel tables 2, 3, 4)                             *
****************************************************************;

Data temp.table;
      set comorbid.cleanedfinal;
run;

***Demo Table1;
           ***Character variables;
proc freq data=temp.table;
      table prescomorb * (acslevel acspedlev edgcseye edgcsmotor
      edgcsverb gender injurytype payment race region sceneeye
```

```
      scenegcsaq scenemotor scenever statelevel tempscale
      YOADMIT)/chisq;
run;

            ***Continous Variables;
proc means data=temp.table n mean std min max nmiss;
      var acs_edrts acs_ps age edbasedef edgcstotal edrtrs
      edsysbp edtemp iss scenetotal triss_prob;
run;

proc ttest data=temp.table;
      class prescomorb;
      var acs_edrts acs_ps age edbasedef edgcstotal edrtrs
      edsysbp edtemp iss scenetotal triss_prob;
      title 'Testing the Equality of Means for Patients with
      Comordities and without';
run;

***Outcome Table2;
proc sort data=temp.table;
      by agecat;
run;

            ***Character variables;
proc freq data=temp.table;
      table agecat * (disstatus Fimexpress Fimfeed
      Fimlocomt)/chisq;
run;

            ***Continous Variables;
proc means data=temp.table n mean std nmiss min max;
      var ventdays Fimtotal ICUDays Los;
run;

proc ttest data=temp.table;
      *by agecat;
      var ventdays Fimtotal ICUDays Los;
      title 'Testing the Equality of Means for Patients in
      Different Age Categories';
run;

***Comorbidity Table3;
proc means data= temp.table n;
      by agecat;
       var acqsum alchsum alzhsum anemsum arthsum ashsum bilsum
      bowelsum brainsum chemosum cirrsum corartsum corpsum
      coudsum cvasum demsum dialsum dmysum drugsum gasvsum
      heartsum hemosum hivsum hpytsum htycardsum insldepsum
      lupussum metasum msclsum ninsldepsum obplsum obstsum
      parksum pepusum ppulsum pregsum psychsum pulcsum seizsum
      sercsum spinalsum sterdsum thpysum transpsum;
run;
```

```
proc freq data=temp.table;
     table agecat * (acqsum alchsum alzhsum anemsum arthsum
     ashsum bilsum bowelsum brainsum chemosum cirrsum corartsum
     corpsum coudsum cvasum demsum dialsum dmysum drugsum
     gasvsum heartsum hemosum hivsum hpytsum htycardsum
     insldepsum lupussum metasum msclsum ninsldepsum obplsum
     obstsum parksum pepusum ppulsum pregsum psychsum pulcsum
     seizsum sercsum spinalsum sterdsum thpysum transpsum
     prescomorb)/chisq;
run;


*****************************************************************
*    ANALYSIS PART 6:  Model Building and Multiple Imputation  *
*                       Using NRI macro (Excel Table 4)          *
*                      Macro code provided in Appendix B        *
*****************************************************************;


title " ";

***********************************************
MODEL-1(unadjusted):  dead = comorbidity
***********************************************;

***acqsum;
proc logistic descending data=comorbid.cleanedfinal;
     model dead= acqsum;
     ods output ParameterEstimates=pe ;
run;

***alchsum;
proc logistic descending data=comorbid.cleanedfinal;
     model dead= alchsum;
     ods output ParameterEstimates=pe ;
run;

***alzhsum;
proc logistic descending data=comorbid.cleanedfinal;
     model dead= alzhsum;
     ods output ParameterEstimates=pe ;
run;

***anemsum;
proc logistic descending data=comorbid.cleanedfinal;
     model dead= anemsum;
     ods output ParameterEstimates=pe ;
run;

***arthsum;
proc logistic descending data=comorbid.cleanedfinal;
```

```
      model dead= arthsum;
      ods output ParameterEstimates=pe ;
run;

***ashsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= ashsum;
      ods output ParameterEstimates=pe ;
run;

***bilsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= bilsum;
      ods output ParameterEstimates=pe ;
run;

***bowelsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= bowelsum;
      ods output ParameterEstimates=pe ;
run;

***brainsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= brainsum;
      ods output ParameterEstimates=pe ;
run;

***chemosum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= chemosum;
      ods output ParameterEstimates=pe ;
run;

***cirrsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= cirrsum;
      ods output ParameterEstimates=pe ;
run;

***corartsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= corartsum;
      ods output ParameterEstimates=pe ;
run;

***corpsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= corpsum;
      ods output ParameterEstimates=pe ;
run;
```

```
***coudsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= coudsum;
      ods output ParameterEstimates=pe ;
run;

***cvasum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= cvasum;
      ods output ParameterEstimates=pe ;
run;

***demsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= demsum;
      ods output ParameterEstimates=pe ;
run;

***dialsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= dialsum;
      ods output ParameterEstimates=pe ;
run;

***dmysum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= dmysum;
      ods output ParameterEstimates=pe ;
run;

***drugsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= drugsum;
      ods output ParameterEstimates=pe ;
run;

***gasvsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= gasvsum;
      ods output ParameterEstimates=pe ;
run;

***heartsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= heartsum;
      ods output ParameterEstimates=pe ;
run;

***hemosum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= hemosum;
      ods output ParameterEstimates=pe ;
```

```
run;

***hivsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= hivsum;
      ods output ParameterEstimates=pe ;
run;

***hpytsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= hpytsum;
      ods output ParameterEstimates=pe ;
run;

***htycardsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= htycardsum;
      ods output ParameterEstimates=pe ;
run;

***insldepsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= insldepsum;
      ods output ParameterEstimates=pe ;
run;

***lupussum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= lupussum;
      ods output ParameterEstimates=pe ;
run;

***metasum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= metasum;
      ods output ParameterEstimates=pe ;
run;

***msclsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= msclsum;
      ods output ParameterEstimates=pe ;
run;

***ninsldepsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= ninsldepsum;
      ods output ParameterEstimates=pe ;
run;

***obplsum;
proc logistic descending data=comorbid.cleanedfinal;
```

```
      model dead= obplsum;
      ods output ParameterEstimates=pe ;
run;

***obstsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= obstsum;
      ods output ParameterEstimates=pe ;
run;

***parksum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= parksum;
      ods output ParameterEstimates=pe ;
run;

***pepusum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= pepusum;
      ods output ParameterEstimates=pe ;
run;

***ppulsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= ppulsum;
      ods output ParameterEstimates=pe ;
run;

***pregsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= pregsum;
      ods output ParameterEstimates=pe ;
run;

***psychsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= psychsum;
      ods output ParameterEstimates=pe ;
run;

***pulcsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= pulcsum;
      ods output ParameterEstimates=pe ;
run;

***seizsum;
proc logistic descending data=comorbid.cleanedfinal;
      model dead= seizsum;
      ods output ParameterEstimates=pe ;
run;
```

```
***sercsum;
proc logistic descending data=comorbid.cleanedfinal;
     model dead= sercsum;
     ods output ParameterEstimates=pe ;
run;

***spinalsum;
proc logistic descending data=comorbid.cleanedfinal;
     model dead= spinalsum;
     ods output ParameterEstimates=pe ;
run;

***sterdsum;
proc logistic descending data=comorbid.cleanedfinal;
     model dead= sterdsum;
     ods output ParameterEstimates=pe ;
run;

***thpysum;
proc logistic descending data=comorbid.cleanedfinal;
     model dead= thpysum;
     ods output ParameterEstimates=pe ;
run;

***transpsum;
proc logistic descending data=comorbid.cleanedfinal;
     model dead= transpsum;
     ods output ParameterEstimates=pe ;
run;

***total;
proc logistic descending data=comorbid.cleanedfinal;
     model dead= prescomorb;
     ods output ParameterEstimates=pe ;
run;


***********************************************
MODEL-2: dead = comorbidity+ triss_prob
***********************************************;

*****************************************
Multiple imputation   Code - model 2
*****************************************;

***acqsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss acqsum;
run;
```

```
proc logistic descending data=miout;
      by _imputation_;
      model dead = acqsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept acqsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***alchsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss alchsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = alchsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept alchsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
```

```
run;

proc print data=pe;
run;

***alzhsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss alzhsum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = alzhsum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
     modeleffects intercept alzhsum triss_prob;
     ods output ParameterEstimates=pe;
run;

data pe;
     set pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***anemsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss anemsum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = anemsum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
```

```
      modeleffects intercept anemsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***arthsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss arthsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = arthsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept arthsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***ashsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
```

```
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss ashsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = ashsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept ashsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***bilsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss bilsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = bilsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept bilsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
```

```
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***bowelsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss bowelsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = bowelsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept bowelsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***brainsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss brainsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = brainsum triss_prob / covb;
```

```
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept brainsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***chemosum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss chemosum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = chemosum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept chemosum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;
```

```
****cirrsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss cirrsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = cirrsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept cirrsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****corartsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss corartsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = corartsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept corartsum triss_prob;
      ods output ParameterEstimates=pe;
run;
```

```
data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****corpsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss corpsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = corpsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept corpsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****coudsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss coudsum;
run;
```

```
proc logistic descending data=miout;
      by _imputation_;
      model dead = coudsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept coudsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****cvasum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss cvasum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = cvasum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept cvasum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
```

```
run;

proc print data=pe;
run;

****demsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss demsum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = demsum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
     modeleffects intercept demsum triss_prob;
     ods output ParameterEstimates=pe;
run;

data pe;
     set pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****dialsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss dialsum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = dialsum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
```

```
      modeleffects intercept dialsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****dmysum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss dmysum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = dmysum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept dmysum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****drugsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
```

```
    ***added in ... because they were low in missing values;
    var dead triss_prob los icudays ventdays edgcstotal edrtrs
    iss drugsum;
run;

proc logistic descending data=miout;
    by _imputation_;
    model dead = drugsum triss_prob / covb;
    ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
    modeleffects intercept drugsum triss_prob;
    ods output ParameterEstimates=pe;
run;

data pe;
    set pe;
    EFFECT=PARM;
    AOR=2.71828**(estimate);
    LOWER=2.71828**(LCLmean);
    UPPER=2.71828**(UCLmean);
    PVALUE=probt;
    keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****gasvsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
    ***added in ... because they were low in missing values;
    var dead triss_prob los icudays ventdays edgcstotal edrtrs
    iss gasvsum;
run;

proc logistic descending data=miout;
    by _imputation_;
    model dead = gasvsum triss_prob / covb;
    ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
    modeleffects intercept gasvsum triss_prob;
    ods output ParameterEstimates=pe;
run;

data pe;
    set pe;
    EFFECT=PARM;
    AOR=2.71828**(estimate);
```

```
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****heartsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss heartsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = heartsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept heartsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****hemosum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss hemosum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = hemosum triss_prob / covb;
```

```
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
     modeleffects intercept hemosum triss_prob;
     ods output ParameterEstimates=pe;
run;

data pe;
     set pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****hivsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss hivsum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = hivsum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
     modeleffects intercept hivsum triss_prob;
     ods output ParameterEstimates=pe;
run;

data pe;
     set pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;
```

```
****hpytsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss hpytsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = hpytsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept hpytsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****htycardsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss htycardsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = htycardsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept htycardsum triss_prob;
      ods output ParameterEstimates=pe;
run;
```

```
data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****insldepsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss insldepsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = insldepsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept insldepsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****lupussum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss lupussum;
run;
```

```
proc logistic descending data=miout;
      by _imputation_;
      model dead = lupussum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept lupussum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

****metasum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss metasum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = metasum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept metasum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
```

```
run;

proc print data=pe;
run;

***msclsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss msclsum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = msclsum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
     modeleffects intercept msclsum triss_prob;
     ods output ParameterEstimates=pe;
run;

data pe;
     set pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***ninsldepsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss ninsldepsum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = ninsldepsum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
```

```
      modeleffects intercept ninsldepsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***obplsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss obplsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = obplsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept obplsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***obstsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
```

```
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss obstsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = obstsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept obstsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***parksum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss parksum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = parksum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept parksum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
```

```
        LOWER=2.71828**(LCLmean);
        UPPER=2.71828**(UCLmean);
        PVALUE=probt;
        keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***pepusum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
        ***added in ... because they were low in missing values;
        var dead triss_prob los icudays ventdays edgcstotal edrtrs
        iss pepusum;
run;

proc logistic descending data=miout;
        by _imputation_;
        model dead = pepusum triss_prob / covb;
        ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
        modeleffects intercept pepusum triss_prob;
        ods output ParameterEstimates=pe;
run;

data pe;
        set pe;
        EFFECT=PARM;
        AOR=2.71828**(estimate);
        LOWER=2.71828**(LCLmean);
        UPPER=2.71828**(UCLmean);
        PVALUE=probt;
        keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***ppulsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
        ***added in ... because they were low in missing values;
        var dead triss_prob los icudays ventdays edgcstotal edrtrs
        iss ppulsum;
run;

proc logistic descending data=miout;
        by _imputation_;
        model dead = ppulsum triss_prob / covb;
```

```
    ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
    modeleffects intercept ppulsum triss_prob;
    ods output ParameterEstimates=pe;
run;

data pe;
    set pe;
    EFFECT=PARM;
    AOR=2.71828**(estimate);
    LOWER=2.71828**(LCLmean);
    UPPER=2.71828**(UCLmean);
    PVALUE=probt;
    keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***pregsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
    ***added in ... because they were low in missing values;
    var dead triss_prob los icudays ventdays edgcstotal edrtrs
    iss pregsum;
run;

proc logistic descending data=miout;
    by _imputation_;
    model dead = pregsum triss_prob / covb;
    ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
    modeleffects intercept pregsum triss_prob;
    ods output ParameterEstimates=pe;
run;

data pe;
    set pe;
    EFFECT=PARM;
    AOR=2.71828**(estimate);
    LOWER=2.71828**(LCLmean);
    UPPER=2.71828**(UCLmean);
    PVALUE=probt;
    keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;
```

```
***psychsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss psychsum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = psychsum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
     modeleffects intercept psychsum triss_prob;
     ods output ParameterEstimates=pe;
run;

data pe;
     set pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***pulcsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss pulcsum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = pulcsum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
     modeleffects intercept pulcsum triss_prob;
     ods output ParameterEstimates=pe;
run;
```

```
data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***seizsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss seizsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = seizsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept seizsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***sercsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss sercsum;
run;
```

```
proc logistic descending data=miout;
      by _imputation_;
      model dead = sercsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept sercsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***spinalsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss spinalsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = spinalsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept spinalsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
```

```
run;

proc print data=pe;
run;

***sterdsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss sterdsum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = sterdsum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
     modeleffects intercept sterdsum triss_prob;
     ods output ParameterEstimates=pe;
run;

data pe;
     set pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***thpysum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
     ***added in ... because they were low in missing values;
     var dead triss_prob los icudays ventdays edgcstotal edrtrs
     iss thpysum;
run;

proc logistic descending data=miout;
     by _imputation_;
     model dead = thpysum triss_prob / covb;
     ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
```

```
      modeleffects intercept thpysum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***transpsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss transpsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = transpsum triss_prob / covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept transpsum triss_prob;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***total;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
```

```
      ***added in ... because they were low in missing values;
      var dead triss_prob los icudays ventdays edgcstotal edrtrs
      iss prescomorb;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = prescomorb triss_prob / covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept prescomorb triss_prob;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;


************************************************
NRI - model 2 - (MACRO INCLUDED IN APPENDIX B)
************************************************;
%include "D:\Practicum\SAS_Macro\rocplus.sas";

***acqsum;
title 'acqsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = acqsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***alchsum;
title 'alchsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = alchsum,
```

```
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***alzhsum;
title 'alzhsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = alzhsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***anemsum;
title 'anemsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = anemsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***arthsum;
title 'arthsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = arthsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***ashsum;
title 'ashsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = ashsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***bilsum;
title 'bilsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = bilsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
```

```
      plots=Y);
run;

***bowelsum;
title 'bowelsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = bowelsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***brainsum;
title 'brainsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = brainsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***chemosum;
title 'chemosum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = chemosum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***cirrsum;
title 'cirrsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = cirrsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***corartsum;
title 'corartsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = corartsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;
```

```
***corpsum;
title 'corpsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = corpsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***coudsum;
title 'coudsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = coudsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***cvasum;
title 'cvasum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = cvasum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***demsum;
title 'demsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = demsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***dialsum;
title 'dialsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = dialsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***dmysum;
```

```
title 'dmysum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = dmysum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***drugsum;
title 'drugsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = drugsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***gasvsum;
title 'gasvsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = gasvsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***heartsum;
title 'heartsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = heartsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***hemosum;
title 'hemosum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = hemosum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***hivsum;
title 'hivsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
```

```
      oldvars = triss_prob,
      addedvars = hivsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***hpytsum;
title 'hpytsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = hpytsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***htycardsum;
title 'htycardsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = htycardsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***insldepsum;
title 'insldepsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = insldepsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***lupussum;
title 'lupussum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = lupussum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***metasum;
title 'metasum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = metasum,
```

```
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***msclsum;
title 'msclsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = msclsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***ninsldepsum;
title 'ninsldepsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = ninsldepsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***obplsum;
title 'obplsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = obplsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***obstsum;
title 'obstsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = obstsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***parksum;
title 'parksum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = parksum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
```

```
      plots=Y);
run;

***pepusum;
title 'pepusum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = pepusum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***ppulsum;
title 'ppulsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = ppulsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***pregsum;
title 'pregsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = pregsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***psychsum;
title 'psychsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = psychsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***pulcsum;
title 'pulcsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = pulcsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;
```

```
***seizsum;
title 'seizsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = seizsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***sercsum;
title 'sercsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = sercsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***spinalsum;
title 'spinalsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = spinalsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***sterdsum;
title 'sterdsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = sterdsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***thpysum;
title 'thpysum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = thpysum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***transpsum;
```

```
title 'transpsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = transpsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***total;
title 'total';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = prescomorb,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***********************************************
MODEL-3: dead = comorbidity+ triss_prob + age
***********************************************;

*****************************************
Multiple imputation   Code - model 3
*****************************************;

***acqsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss acqsum;
run;

proc logistic descending data=miout;
      by _imputation_;
      model dead = acqsum triss_prob age/ covb;
      ods output ParameterEstimates=lgsparms covb=lgscovb;
run;

proc mianalyze parms=lgsparms covb(effectvar=stacking)=lgscovb;
      modeleffects intercept acqsum triss_prob age;
      ods output ParameterEstimates=pe;
run;

data pe;
      set pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
```

```
    PVALUE=probt;
    keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=pe;
run;

***alchsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
    ***added in ... because they were low in missing values;
    var dead age triss_prob los icudays ventdays edgcstotal
    edrtrs iss alchsum;
run;

proc logistic descending data=temp.miout;
    by _imputation_;
    model dead = alchsum triss_prob age/ covb;
    ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
    modeleffects intercept alchsum triss_prob age;
    ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
    set temp.pe;
    EFFECT=PARM;
    AOR=2.71828**(estimate);
    LOWER=2.71828**(LCLmean);
    UPPER=2.71828**(UCLmean);
    PVALUE=probt;
    keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***alzhsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
    ***added in ... because they were low in missing values;
    var dead age triss_prob los icudays ventdays edgcstotal
    edrtrs iss alzhsum;
run;

proc logistic descending data=temp.miout;
    by _imputation_;
    model dead = alzhsum triss_prob age/ covb;
```

```
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept alzhsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***anemsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss anemsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = anemsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept anemsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
```

```
run;

proc print data=temp.pe;
run;

***arthsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss arthsum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = arthsum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
     covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
     modeleffects intercept arthsum triss_prob age;
     ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
     set temp.pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***ashsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss ashsum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = ashsum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
     covb=temp.lgscovb;
```

```
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept ashsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***bilsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss bilsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = bilsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
      covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept bilsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;
```

```
proc print data=temp.pe;
run;

***bowelsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss bowelsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = bowelsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept bowelsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***brainsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss brainsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = brainsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;
```

```
proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept brainsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***chemosum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss chemosum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = chemosum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept chemosum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;
```

```
****cirrsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss cirrsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = cirrsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept cirrsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****corartsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss corartsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = corartsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
```

```
      modeleffects intercept corartsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****corpsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss corpsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = corpsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept corpsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****coudsum;
```

```
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss coudsum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = coudsum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
     modeleffects intercept coudsum triss_prob age;
     ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
     set temp.pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****cvasum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss cvasum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = cvasum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
     modeleffects intercept cvasum triss_prob age;
     ods output ParameterEstimates=temp.pe;
```

```
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****demsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss demsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = demsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept demsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****dialsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
```

```
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss dialsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = dialsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept dialsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****dmysum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss dmysum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = dmysum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept dmysum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;
```

```
data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****drugsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss drugsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = drugsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept drugsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****gasvsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
```

```
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss gasvsum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = gasvsum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
     modeleffects intercept gasvsum triss_prob age;
     ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
     set temp.pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****heartsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss heartsum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = heartsum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
     modeleffects intercept heartsum triss_prob age;
     ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
```

```
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****hemosum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss hemosum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = hemosum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept hemosum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****hivsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss hivsum;
```

```
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = hivsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept hivsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****hpytsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss hpytsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = hpytsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept hpytsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
```

```
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****htycardsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss htycardsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = htycardsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept htycardsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****insldepsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss insldepsum;
run;
```

```
proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = insldepsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept insldepsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****lupussum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss lupussum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = lupussum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept lupussum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
```

```
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

****metasum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss metasum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = metasum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept metasum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***msclsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss msclsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
```

```
      model dead = msclsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept msclsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***ninsldepsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss ninsldepsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = ninsldepsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept ninsldepsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
```

```
    keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***obplsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
    ***added in ... because they were low in missing values;
    var dead age triss_prob los icudays ventdays edgcstotal
    edrtrs iss obplsum;
run;

proc logistic descending data=temp.miout;
    by _imputation_;
    model dead = obplsum triss_prob age/ covb;
    ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
    modeleffects intercept obplsum triss_prob age;
    ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
    set temp.pe;
    EFFECT=PARM;
    AOR=2.71828**(estimate);
    LOWER=2.71828**(LCLmean);
    UPPER=2.71828**(UCLmean);
    PVALUE=probt;
    keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***obstsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
    ***added in ... because they were low in missing values;
    var dead age triss_prob los icudays ventdays edgcstotal
    edrtrs iss obstsum;
run;

proc logistic descending data=temp.miout;
    by _imputation_;
    model dead = obstsum triss_prob age/ covb;
```

```
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept obstsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***parksum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss parksum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = parksum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept parksum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
```

```
run;

proc print data=temp.pe;
run;

***pepusum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss pepusum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = pepusum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
     modeleffects intercept pepusum triss_prob age;
     ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
     set temp.pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***ppulsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss ppulsum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = ppulsum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
```

```
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept ppulsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***pregsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss pregsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = pregsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept pregsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;
```

```
proc print data=temp.pe;
run;

***psychsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss psychsum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = psychsum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
     modeleffects intercept psychsum triss_prob age;
     ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
     set temp.pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***pulcsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss pulcsum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = pulcsum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;
```

```
proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept pulcsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***seizsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss seizsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = seizsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept seizsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;
```

```
***sercsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss sercsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = sercsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept sercsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***spinalsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss spinalsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = spinalsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
```

```
      modeleffects intercept spinalsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***sterdsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss sterdsum;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = sterdsum triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept sterdsum triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***thpysum;
```

```
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss thpysum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = thpysum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
     modeleffects intercept thpysum triss_prob age;
     ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
     set temp.pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***transpsum;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss transpsum;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = transpsum triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
     modeleffects intercept transpsum triss_prob age;
     ods output ParameterEstimates=temp.pe;
```

```
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***total;
proc mi data=comorbid.cleanedfinal noprint nimpute=5
seed=05262007 out=temp.miout;
      ***added in ... because they were low in missing values;
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss prescomorb;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = prescomorb triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept prescomorb triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

**********************************************
NRI - model 3- (MACRO INCLUDED IN APPENDIX B)
**********************************************;
```

```
%include "D:\Practicum\SAS_Macro\rocplus.sas";

***acqsum;
title 'acqsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = acqsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***alchsum;
title 'alchsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = alchsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***alzhsum;
title 'alzhsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = alzhsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***anemsum;
title 'anemsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = anemsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***arthsum;
title 'arthsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = arthsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;
```

```
***ashsum;
title 'ashsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = ashsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***bilsum;
title 'bilsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = bilsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***bowelsum;
title 'bowelsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = bowelsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***brainsum;
title 'brainsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = brainsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***chemosum;
title 'chemosum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = chemosum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***cirrsum;
title 'cirrsum';
```

```
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = cirrsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***corartsum;
title 'corartsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = corartsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***corpsum;
title 'corpsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = corpsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***coudsum;
title 'coudsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = coudsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***cvasum;
title 'cvasum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = cvasum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***demsum;
title 'demsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
```

```
      addedvars = demsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***dialsum;
title 'dialsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = dialsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***dmysum;
title 'dmysum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = dmysum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***drugsum;
title 'drugsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = drugsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***gasvsum;
title 'gasvsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = gasvsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***heartsum;
title 'heartsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = heartsum,
      risk= .01 .02 .03 .05,
```

```
      idvar = inc_key,
      plots=Y);
run;

***hemosum;
title 'hemosum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = hemosum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***hivsum;
title 'hivsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = hivsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***hpytsum;
title 'hpytsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = hpytsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***htycardsum;
title 'htycardsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = htycardsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***insldepsum;
title 'insldepsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = insldepsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
```

```
run;

***lupussum;
title 'lupussum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = lupussum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***metasum;
title 'metasum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = metasum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***msclsum;
title 'msclsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = msclsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***ninsldepsum;
title 'ninsldepsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = ninsldepsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***obplsum;
title 'obplsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = obplsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;
```

```
***obstsum;
title 'obstsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = obstsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***parksum;
title 'parksum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = parksum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***pepusum;
title 'pepusum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = pepusum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***ppulsum;
title 'ppulsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = ppulsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***pregsum;
title 'pregsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = pregsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***psychsum;
title 'psychsum';
```

```
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = psychsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***pulcsum;
title 'pulcsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = pulcsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***seizsum;
title 'seizsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = seizsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***sercsum;
title 'sercsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = sercsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***spinalsum;
title 'spinalsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = spinalsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***sterdsum;
title 'sterdsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
```

```
      addedvars = sterdsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***thpysum;
title 'thpysum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = thpysum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***transpsum;
title 'transpsum';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob age,
      addedvars = transpsum,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***total;
title 'total';
%rocplus(data=comorbid.cleanedfinal, event=dead,
      oldvars = triss_prob,
      addedvars = prescomorb,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;


*****************************************************************
*     ANALYSIS PART 7:  ICUDays, Ventdays, Los Models           *
*                       (Excel Tables 5, 6, 7)                  *
*****************************************************************;

***********************************************
MODEL: Los = comorbidity + triss_prob
***********************************************;
data temp;
      set comorbid.cleanedfinal;
run;

***acqsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob acqsum / clb;
```

```
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class acqsum;
      var Los;
run;

***alchsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob alchsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class alchsum;
      var Los;
run;

***alzhsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob alzhsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class alzhsum;
      var Los;
run;

***anemsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob anemsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class anemsum;
      var Los;
run;

***arthsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob arthsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class arthsum;
      var Los;
run;
```

```
***ashsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob ashsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class ashsum;
      var Los;
run;

***bilsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob bilsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class bilsum;
      var Los;
run;

***bowelsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob bowelsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class bowelsum;
      var Los;
run;

***brainsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob brainsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class brainsum;
      var Los;
run;

***chemosum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob chemosum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class chemosum;
```

```
      var Los;
run;

***cirrsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob cirrsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class cirrsum;
      var Los;
run;

***corartsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob corartsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class corartsum;
      var Los;
run;

***corpsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob corpsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class corpsum;
      var Los;
run;

***coudsum;
 proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob coudsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class coudsum;
      var Los;
run;

***cvasum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob cvasum / clb;
      ods output ParameterEstimates=pe ;
run;
```

```
proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class cvasum;
      var Los;
run;

***demsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob demsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class demsum;
      var Los;
run;

***dialsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob dialsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class dialsum;
      var Los;
run;

***dmysum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob dmysum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class dmysum;
      var Los;
run;

***drugsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob drugsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class drugsum;
      var Los;
run;

***gasvsum;
proc reg data=temp; *comorbid.cleanedfinal;
```

```
      model Los= triss_prob gasvsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class gasvsum;
      var Los;
run;

***heartsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob heartsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class heartsum;
      var Los;
run;

***hemosum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob hemosum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class hemosum;
      var Los;
run;

***hivsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob hivsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class hivsum;
      var Los;
run;

***hpytsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob hpytsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class hpytsum;
      var Los;
run;
```

```
***htycardsum;
 proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob htycardsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class htycardsum;
      var Los;
run;

***insldepsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob insldepsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class insldepsum;
      var Los;
run;

***lupussum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob lupussum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class lupussum;
      var Los;
run;

***metasum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob metasum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class metasum;
      var Los;
run;

***msclsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob msclsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
```

```
      class msclsum;
      var Los;
run;

***ninsldepsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob ninsldepsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class ninsldepsum;
      var Los;
run;

***obplsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob obplsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class obplsum;
      var Los;
run;

***obstsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob obstsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class obstsum;
      var Los;
run;

***parksum;
 proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob parksum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class parksum;
      var Los;
run;

***pepusum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob pepusum / clb;
      ods output ParameterEstimates=pe ;
```

```
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class pepusum;
      var Los;
run;

***ppulsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob ppulsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class ppulsum;
      var Los;
run;

***pregsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob pregsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class pregsum;
      var Los;
run;

***psychsum;
 proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob psychsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class psychsum;
      var Los;
run;

***pulcsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob pulcsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class pulcsum;
      var Los;
run;

***seizsum;
```

```
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob seizsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class seizsum;
      var Los;
run;

***sercsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob sercsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class sercsum;
      var Los;
run;

***sterdsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob sterdsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class sterdsum;
      var Los;
run;

***thpysum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob thpysum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class thpysum;
      var Los;
run;

***transpsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Los= triss_prob transpsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class transpsum;
      var Los;
```

```
run;

***total;
proc reg data=temp; *comorbid.cleanedfinal;
     model Los= triss_prob prescomorb / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class prescomorb;
     var Los;
run;

***********************************************
MODEL: ICUDays = comorbidity + triss_prob
***********************************************;

***acqsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob acqsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class acqsum;
     var ICUDays;
run;

***alchsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob alchsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class alchsum;
     var ICUDays;
run;

***alzhsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob alzhsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class alzhsum;
     var ICUDays;
run;

***anemsum;
proc reg data=temp; *comorbid.cleanedfinal;
```

```
        model ICUDays= triss_prob anemsum / clb;
        ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
        class anemsum;
        var ICUDays;
run;

***arthsum;
proc reg data=temp; *comorbid.cleanedfinal;
        model ICUDays= triss_prob arthsum / clb;
        ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
        class arthsum;
        var ICUDays;
run;

***ashsum;
proc reg data=temp; *comorbid.cleanedfinal;
        model ICUDays= triss_prob ashsum / clb;
        ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
        class ashsum;
        var ICUDays;
run;

***bilsum;
proc reg data=temp; *comorbid.cleanedfinal;
        model ICUDays= triss_prob bilsum / clb;
        ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
        class bilsum;
        var ICUDays;
run;

***bowelsum;
proc reg data=temp; *comorbid.cleanedfinal;
        model ICUDays= triss_prob bowelsum / clb;
        ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
        class bowelsum;
        var ICUDays;
run;
```

```
***brainsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob brainsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class brainsum;
      var ICUDays;
run;

***chemosum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob chemosum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class chemosum;
      var ICUDays;
run;

***cirrsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob cirrsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class cirrsum;
      var ICUDays;
run;

***corartsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob corartsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class corartsum;
      var ICUDays;
run;

***corpsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob corpsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
```

```
      class corpsum;
      var ICUDays;
run;

***coudsum;
 proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob coudsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class coudsum;
      var ICUDays;
run;

***cvasum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob cvasum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class cvasum;
      var ICUDays;
run;

***demsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob demsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class demsum;
      var ICUDays;
run;

***dialsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob dialsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class dialsum;
      var ICUDays;
run;

***dmysum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob dmysum / clb;
      ods output ParameterEstimates=pe ;
```

```
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class dmysum;
      var ICUDays;
run;

***drugsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob drugsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class drugsum;
      var ICUDays;
run;

***gasvsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob gasvsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class gasvsum;
      var ICUDays;
run;

***heartsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob heartsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class heartsum;
      var ICUDays;
run;

***hemosum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob hemosum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class hemosum;
      var ICUDays;
run;

***hivsum;
```

```
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob hivsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class hivsum;
      var ICUDays;
run;

***hpytsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob hpytsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class hpytsum;
      var ICUDays;
run;

***htycardsum;
 proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob htycardsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class htycardsum;
      var ICUDays;
run;

***insldepsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob insldepsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class insldepsum;
      var ICUDays;
run;

***lupussum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob lupussum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class lupussum;
      var ICUDays;
```

```
run;

***metasum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob metasum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class metasum;
     var ICUDays;
run;

***msclsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob msclsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class msclsum;
     var ICUDays;
run;

***ninsldepsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob ninsldepsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class ninsldepsum;
     var ICUDays;
run;

***obplsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob obplsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class obplsum;
     var ICUDays;
run;

***obstsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob obstsum / clb;
     ods output ParameterEstimates=pe ;
run;
```

```
proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class obstsum;
     var ICUDays;
run;

***parksum;
 proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob parksum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class parksum;
     var ICUDays;
run;

***pepusum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob pepusum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class pepusum;
     var ICUDays;
run;

***ppulsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob ppulsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class ppulsum;
     var ICUDays;
run;

***pregsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob pregsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class pregsum;
     var ICUDays;
run;

***psychsum;
 proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob psychsum / clb;
```

```
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class psychsum;
      var ICUDays;
run;

***pulcsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob pulcsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class pulcsum;
      var ICUDays;
run;

***seizsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob seizsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class seizsum;
      var ICUDays;
run;

***sercsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob sercsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class sercsum;
      var ICUDays;
run;

***spinalsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model ICUDays= triss_prob spinalsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class spinalsum;
      var ICUDays;
run;
```

```
***sterdsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob sterdsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class sterdsum;
     var ICUDays;
run;

***thpysum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob thpysum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class thpysum;
     var ICUDays;
run;

***transpsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob transpsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class transpsum;
     var ICUDays;
run;

***total;
proc reg data=temp; *comorbid.cleanedfinal;
     model ICUDays= triss_prob prescomorb / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class prescomorb;
     var ICUDays;
run;

***********************************************
MODEL: Ventdays = comorbidity + triss_prob
***********************************************;

***acqsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model Ventdays= triss_prob acqsum / clb;
     ods output ParameterEstimates=pe ;
```

```
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class acqsum;
      var Ventdays;
run;

***alchsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob alchsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class alchsum;
      var Ventdays;
run;

***alzhsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob alzhsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class alzhsum;
      var Ventdays;
run;

***anemsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob anemsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class anemsum;
      var Ventdays;
run;

***arthsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob arthsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class arthsum;
      var Ventdays;
run;

***ashsum;
```

```
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob ashsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class ashsum;
      var Ventdays;
run;

***bilsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob bilsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class bilsum;
      var Ventdays;
run;

***bowelsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob bowelsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class bowelsum;
      var Ventdays;
run;

***brainsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob brainsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class brainsum;
      var Ventdays;
run;

***chemosum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob chemosum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class chemosum;
      var Ventdays;
```

```
run;

***cirrsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model Ventdays= triss_prob cirrsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class cirrsum;
     var Ventdays;
run;

***corartsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model Ventdays= triss_prob corartsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class corartsum;
     var Ventdays;
run;

***corpsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model Ventdays= triss_prob corpsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class corpsum;
     var Ventdays;
run;

***coudsum;
 proc reg data=temp; *comorbid.cleanedfinal;
     model Ventdays= triss_prob coudsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class coudsum;
     var Ventdays;
run;

***cvasum;
proc reg data=temp; *comorbid.cleanedfinal;
     model Ventdays= triss_prob cvasum / clb;
     ods output ParameterEstimates=pe ;
run;
```

```
proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class cvasum;
      var Ventdays;
run;

***demsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob demsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class demsum;
      var Ventdays;
run;

***dialsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob dialsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class dialsum;
      var Ventdays;
run;

***dmysum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob dmysum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class dmysum;
      var Ventdays;
run;

***drugsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob drugsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class drugsum;
      var Ventdays;
run;

***gasvsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob gasvsum / clb;
```

```
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class gasvsum;
      var Ventdays;
run;

***heartsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob heartsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class heartsum;
      var Ventdays;
run;

***hemosum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob hemosum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class hemosum;
      var Ventdays;
run;

***hivsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob hivsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class hivsum;
      var Ventdays;
run;

***hpytsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob hpytsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class hpytsum;
      var Ventdays;
run;
```

```
***htycardsum;
 proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob htycardsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class htycardsum;
      var Ventdays;
run;

***insldepsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob insldepsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class insldepsum;
      var Ventdays;
run;

***lupussum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob lupussum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class lupussum;
      var Ventdays;
run;

***metasum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob metasum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class metasum;
      var Ventdays;
run;

***msclsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob msclsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class msclsum;
```

```
      var Ventdays;
run;

***ninsldepsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob ninsldepsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class ninsldepsum;
      var Ventdays;
run;

***obplsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob obplsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class obplsum;
      var Ventdays;
run;

***obstsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob obstsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class obstsum;
      var Ventdays;
run;

***parksum;
 proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob parksum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class parksum;
      var Ventdays;
run;

***pepusum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob pepusum / clb;
      ods output ParameterEstimates=pe ;
run;
```

```
proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class pepusum;
     var Ventdays;
run;

***ppulsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model Ventdays= triss_prob ppulsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class ppulsum;
     var Ventdays;
run;

***pregsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model Ventdays= triss_prob pregsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class pregsum;
     var Ventdays;
run;

***psychsum;
 proc reg data=temp; *comorbid.cleanedfinal;
     model Ventdays= triss_prob psychsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class psychsum;
     var Ventdays;
run;

***pulcsum;
proc reg data=temp; *comorbid.cleanedfinal;
     model Ventdays= triss_prob pulcsum / clb;
     ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
     class pulcsum;
     var Ventdays;
run;

***seizsum;
proc reg data=temp; *comorbid.cleanedfinal;
```

```
	model Ventdays= triss_prob seizsum / clb;
	ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
	class seizsum;
	var Ventdays;
run;

***sercsum;
proc reg data=temp; *comorbid.cleanedfinal;
	model Ventdays= triss_prob sercsum / clb;
	ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
	class sercsum;
	var Ventdays;
run;

***spinalsum;
proc reg data=temp; *comorbid.cleanedfinal;
	model Ventdays= triss_prob spinalsum / clb;
	ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
	class spinalsum;
	var Ventdays;
run;

***sterdsum;
proc reg data=temp; *comorbid.cleanedfinal;
	model Ventdays= triss_prob sterdsum / clb;
	ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
	class sterdsum;
	var Ventdays;
run;

***thpysum;
proc reg data=temp; *comorbid.cleanedfinal;
	model Ventdays= triss_prob thpysum / clb;
	ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
	class thpysum;
	var Ventdays;
run;
```

```
***transpsum;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob transpsum / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class transpsum;
      var Ventdays;
run;

***total;
proc reg data=temp; *comorbid.cleanedfinal;
      model Ventdays= triss_prob prescomorb / clb;
      ods output ParameterEstimates=pe ;
run;

proc means data=temp mean std; *comorbid.cleanedfinal mean;
      class prescomorb;
      var Ventdays;
run;


*****************************************************************
*     ANALYSIS PART 8:  Severity Scoring (Excel Table 8)        *
*                       Using ORs from comorbidity model3       *
****************************************************************;

**********************************************
Original ORs Used
*********************************************;

Data comorbid.score;
      set comorbid.cleanedfinal;
      if acqsum=1 then acqcomp=2.890;
           else acqcomp=1;
      if alchsum=1 then alchcomp=0.764;
           else alchcomp=1;
      if alzhsum=1 then alzhcomp=1.026;
           else alzhcomp=1;
      if anemsum=1 then anemcomp=1.319;
           else anemcomp=1;
      if arthsum=1 then arthcomp=0.619;
           else arthcomp=1;
      if ashsum=1 then ashcomp=0.488;
           else ashcomp=1;
      if bilsum=1 then bilcomp=3.049;
           else bilcomp=1;
      if bowelsum=1 then bowelcomp=0.654;
           else bowelcomp=1;
      if brainsum=1 then braincomp=2.209;
```

```
      else braincomp=1;
if chemosum=1 then chemocomp=1.487;
      else chemocomp=1;
if cirrsum=1 then cirrcomp=3.941;
      else cirrcomp=1;
if corartsum=1 then corartcomp=1.403;
      else corartcomp=1;
if corpsum=1 then corpcomp=1.197;
      else corpcomp=1;
if coudsum=1 then coudcomp=1.637;
      else coudcomp=1;
if cvasum=1 then cvacomp=1.191;
      else cvacomp=1;
if demsum=1 then demcomp=0.930;
      else demcomp=1;
if dialsum=1 then dialcomp=2.060;
      else dialcomp=1;
if dmysum=1 then dmycomp=1.424;
      else dmycomp=1;
if drugsum=1 then drugcomp=0.649;
      else drugcomp=1;
if gasvsum=1 then gasvcomp=0.900;
      else gasvcomp=1;
if heartsum=1 then heartcomp=1.494;
      else heartcomp=1;
if hemosum=1 then hemocomp=0.535;
      else hemocomp=1;
if hivsum=1 then hivcomp=1.243;
      else hivcomp=1;
if hpytsum=1 then hpytcomp=0.763;
      else hpytcomp=1;
if htycardsum=1 then htycardcomp=2.822;
      else htycardcomp=1;
if insldepsum=1 then insldepcomp=1.051;
      else insldepcomp=1;
if lupussum=1 then lupuscomp=0.673;
      else lupuscomp=1;
if metasum=1 then metacomp=1.977;
      else metacomp=1;
if msclsum=1 then msclcomp=0.592;
      else msclcomp=1;
if ninsldepsum=1 then ninsldepcomp=0.977;
      else ninsldepcomp=1;
if obplsum=1 then obplcomp=1.351;
      else obplcomp=1;
if obstsum=1 then obstcomp=0.974;
      else obstcomp=1;
if parksum=1 then parkcomp=1.111;
      else parkcomp=1;
if pepusum=1 then pepucomp=0.780;
      else pepucomp=1;
if ppulsum=1 then ppulcomp=1.479;
```

```
            else ppulcomp=1;
        if pregsum=1 then pregcomp=0.458;
            else pregcomp=1;
        if psychsum=1 then psychcomp=0.699;
            else psychcomp=1;
        if pulcsum=1 then pulccomp=0.861;
            else pulccomp=1;
        if seizsum=1 then seizcomp=0.836;
            else seizcomp=1;
        if sercsum=1 then serccomp=2.598;
            else serccomp=1;
        if spinalsum=1 then spinalcomp=2.951;
            else spinalcomp=1;
        if sterdsum=1 then sterdcomp=1.388;
            else sterdcomp=1;
        if thpysum=1 then thpycomp=1.105;
            else thpycomp=1;
        if transpsum=1 then transpcomp=1.541;
            else transpcomp=1;
        comorscore = acqcomp * alchcomp * alzhcomp * anemcomp *
        arthcomp * ashcomp * bilcomp * bowelcomp * braincomp *
        chemocomp * cirrcomp * corartcomp * corpcomp * coudcomp *
        cvacomp * demcomp * dialcomp * dmycomp * drugcomp *
        gasvcomp * heartcomp * hemocomp * hivcomp * hpytcomp *
        htycardcomp * insldepcomp * lupuscomp * metacomp * msclcomp
        * ninsldepcomp * obplcomp * obstcomp * parkcomp * pepucomp
        * ppulcomp * pregcomp * psychcomp * pulccomp * seizcomp *
        serccomp * spinalcomp * sterdcomp * thpycomp * transpcomp;
run;

proc print data= comorbid.score (obs=500);
        var inc_key comorscore acqsum alchsum alzhsum anemsum
        arthsum ashsum bilsum bowelsum brainsum chemosum cirrsum
        corartsum corpsum coudsum cvasum demsum dialsum dmysum
        drugsum gasvsum heartsum hemosum hivsum hpytsum htycardsum
        insldepsum lupussum metasum msclsum ninsldepsum obplsum
        obstsum parksum pepusum ppulsum pregsum psychsum pulcsum
        seizsum sercsum spinalsum sterdsum thpysum transpsum;
run;

***unadjusted model;
proc logistic descending data=comorbid.score;
        model dead = triss_prob age;
run;

***Score Model;
proc logistic descending data=comorbid.score;
        model dead = triss_prob age comorscore;
run;
*Results: OR (95%CI) 1.270 (1.259, 1.281)
```

```
***Score Model with MI;
proc mi data=comorbid.score noprint nimpute=5 seed=05252007
out=temp.miout;
     ***added in ... because they were low in missing values;
     var dead age triss_prob los icudays ventdays edgcstotal
     edrtrs iss comorscore;
run;

proc logistic descending data=temp.miout;
     by _imputation_;
     model dead = comorscore triss_prob age/ covb;
     ods output ParameterEstimates=temp.lgsparms
     covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
     modeleffects intercept comorscore triss_prob age;
     ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
     set temp.pe;
     EFFECT=PARM;
     AOR=2.71828**(estimate);
     LOWER=2.71828**(LCLmean);
     UPPER=2.71828**(UCLmean);
     PVALUE=probt;
     keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;

***NRI and IDI of Score Model;
     *Macro code included as reference in Appendix B;
%include "D:\Practicum\SAS_Macro\rocplus.sas";

%rocplus(data=comorbid.score, event=dead,
     oldvars = triss_prob age,
     addedvars = comorscore,
     risk= .01 .02 .03 .05,
     idvar = inc_key,
     plots=Y);
run;

***comparison model;
proc logistic descending data=comorbid.score;
     model dead = triss_prob age prescomorb;
run;
```

```
***Los, Ventdays, ICUdays models;
proc reg data=comorbid.score;
      model Los = triss_prob comorscore/ clb;
      ods output ParameterEstimates=pe ;
run;
*Results: Beta2 (95% CI) 1.19 (1.16, 1.21);

proc reg data=comorbid.score;
      model ICUDays = triss_prob comorscore/ clb;
      ods output ParameterEstimates=pe ;
run;
*Results: Beta2 (95% CI) 0.74  (0.72, 0.76);

proc reg data=comorbid.score;
      model Ventdays = triss_prob comorscore/ clb;
      ods output ParameterEstimates=pe ;
run;
*Results: Beta2 (95% CI) 0.57 (0.56, 0.59);


*******************************************************
Beneficial ORs Recorded as 1 - alchcomp, arthcomp,
ashcomp, bowelcomp, demcomp, drugcomp, gasvcomp,
hemocomp, hpytcomp, lupuscomp, msclcomp, ninsldepcomp,
obstcomp, pepucomp, pregcomp, psychcomp, pulccomp,
seizcomp
*******************************************************;

Data comorbid.score2;
      set comorbid.cleanedfinal;
      if acqsum=1 then acqcomp2=2.890;
            else acqcomp2=1;
      if alchsum=1 then alchcomp2=1.000;
            else alchcomp2=1;
      if alzhsum=1 then alzhcomp2=1.026;
            else alzhcomp2=1;
      if anemsum=1 then anemcomp2=1.319;
            else anemcomp2=1;
      if arthsum=1 then arthcomp2=1.000;
            else arthcomp2=1;
      if ashsum=1 then ashcomp2=1.000;
            else ashcomp2=1;
      if bilsum=1 then bilcomp2=3.049;
            else bilcomp2=1;
      if bowelsum=1 then bowelcomp2=1.000;
            else bowelcomp2=1;
      if brainsum=1 then braincomp2=2.209;
            else braincomp2=1;
      if chemosum=1 then chemocomp2=1.487;
            else chemocomp2=1;
      if cirrsum=1 then cirrcomp2=3.941;
            else cirrcomp2=1;
```

```
if corartsum=1 then corartcomp2=1.403;
     else corartcomp2=1;
if corpsum=1 then corpcomp2=1.197;
     else corpcomp2=1;
if coudsum=1 then coudcomp2=1.637;
     else coudcomp2=1;
if cvasum=1 then cvacomp2=1.191;
     else cvacomp2=1;
if demsum=1 then demcomp2=1.000;
     else demcomp2=1;
if dialsum=1 then dialcomp2=2.060;
     else dialcomp2=1;
if dmysum=1 then dmycomp2=1.424;
     else dmycomp2=1;
if drugsum=1 then drugcomp2=1.000;
     else drugcomp2=1;
if gasvsum=1 then gasvcomp2=1.000;
     else gasvcomp2=1;
if heartsum=1 then heartcomp2=1.494;
     else heartcomp2=1;
if hemosum=1 then hemocomp2=1.000;
     else hemocomp2=1;
if hivsum=1 then hivcomp2=1.243;
     else hivcomp2=1;
if hpytsum=1 then hpytcomp2=1.000;
     else hpytcomp2=1;
if htycardsum=1 then htycardcomp2=2.822;
     else htycardcomp2=1;
if insldepsum=1 then insldepcomp2=1.051;
     else insldepcomp2=1;
if lupussum=1 then lupuscomp2=1.000;
     else lupuscomp2=1;
if metasum=1 then metacomp2=1.977;
     else metacomp2=1;
if msclsum=1 then msclcomp2=1.000;
     else msclcomp2=1;
if ninsldepsum=1 then ninsldepcomp2=1.000;
     else ninsldepcomp2=1;
if obplsum=1 then obplcomp2=1.351;
     else obplcomp2=1;
if obstsum=1 then obstcomp2=1.000;
     else obstcomp2=1;
if parksum=1 then parkcomp2=1.111;
     else parkcomp2=1;
if pepusum=1 then pepucomp2=1.000;
     else pepucomp2=1;
if ppulsum=1 then ppulcomp2=1.479;
     else ppulcomp2=1;
if pregsum=1 then pregcomp2=1.000;
     else pregcomp2=1;
if psychsum=1 then psychcomp2=1.000;
     else psychcomp2=1;
```

```
      if pulcsum=1 then pulccomp2=1.000;
            else pulccomp2=1;
      if seizsum=1 then seizcomp2=1.000;
            else seizcomp2=1;
      if sercsum=1 then serccomp2=2.598;
            else serccomp2=1;
      if spinalsum=1 then spinalcomp2=2.951;
            else spinalcomp2=1;
      if sterdsum=1 then sterdcomp2=1.388;
            else sterdcomp2=1;
      if thpysum=1 then thpycomp2=1.105;
            else thpycomp2=1;
      if transpsum=1 then transpcomp2=1.541;
            else transpcomp2=1;
      comorscore2 = acqcomp2 * alchcomp2 * alzhcomp2 * anemcomp2
      * arthcomp2 * ashcomp2 * bilcomp2 * bowelcomp2 * braincomp2
      * chemocomp2 * cirrcomp2 * corartcomp2 * corpcomp2 *
      coudcomp2 * cvacomp2 * demcomp2 * dialcomp2 * dmycomp2 *
      drugcomp2 * gasvcomp2 * heartcomp2 * hemocomp2 * hivcomp2 *
      hpytcomp2 * htycardcomp2 * insldepcomp2 * lupuscomp2 *
      metacomp2 * msclcomp2 * ninsldepcomp2 * obplcomp2 *
      obstcomp2 * parkcomp2 * pepucomp2 * ppulcomp2 * pregcomp2 *
      psychcomp2 * pulccomp2 * seizcomp2 * serccomp2 *
      spinalcomp2 * sterdcomp2 * thpycomp2 * transpcomp2;
run;

proc print data= comorbid.score2 (obs=500);
      var inc_key comorscore2 acqsum alchsum alzhsum anemsum
      arthsum ashsum bilsum bowelsum brainsum chemosum cirrsum
      corartsum corpsum coudsum cvasum demsum dialsum dmysum
      drugsum gasvsum heartsum hemosum hivsum hpytsum htycardsum
      insldepsum lupussum metasum msclsum ninsldepsum obplsum
      obstsum parksum pepusum ppulsum pregsum psychsum pulcsum
      seizsum sercsum spinalsum sterdsum thpysum transpsum;
run;

***unadjusted model;
proc logistic descending data=comorbid.score2;
      model dead = triss_prob age;
run;

***Score Model;
proc logistic descending data=comorbid.score2;
      model dead = triss_prob age comorscore2;
run;
*Results: OR (95%CI) 1.182 (1.173, 1.191);

***Score Model with MI;
proc mi data=comorbid.score2 noprint nimpute=5 seed=05252007
out=temp.miout;
      ***added in ... because they were low in missing values;
```

```
      var dead age triss_prob los icudays ventdays edgcstotal
      edrtrs iss comorscore2;
run;

proc logistic descending data=temp.miout;
      by _imputation_;
      model dead = comorscore2 triss_prob age/ covb;
      ods output ParameterEstimates=temp.lgsparms
      covb=temp.lgscovb;
run;

proc mianalyze parms=temp.lgsparms
covb(effectvar=stacking)=temp.lgscovb;
      modeleffects intercept comorscore2 triss_prob age;
      ods output ParameterEstimates=temp.pe;
run;

data temp.pe;
      set temp.pe;
      EFFECT=PARM;
      AOR=2.71828**(estimate);
      LOWER=2.71828**(LCLmean);
      UPPER=2.71828**(UCLmean);
      PVALUE=probt;
      keep EFFECT AOR LOWER UPPER PVALUE;
run;

proc print data=temp.pe;
run;


***NRI and IDI of Score Model;
      *Macro code included as reference in Appendix B;
%include "D:\Practicum\SAS_Macro\rocplus.sas";

%rocplus(data=comorbid.score2, event=dead,
      oldvars = triss_prob age,
      addedvars = comorscore2,
      risk= .01 .02 .03 .05,
      idvar = inc_key,
      plots=Y);
run;

***comparison model;
proc logistic descending data=comorbid.score2;
      model dead = triss_prob age prescomorb;
run;

***Los, Ventdays, ICUdays models;
proc reg data=comorbid.score2;
      model Los = triss_prob comorscore2/ clb;
      ods output ParameterEstimates=pe ;
```

```
run;
*Results: Beta2 (95% CI) ;

proc reg data=comorbid.score2;
     model ICUDays = triss_prob comorscore2/ clb;
     ods output ParameterEstimates=pe ;
run;
*Results: Beta2 (95% CI) ;

proc reg data=comorbid.score2;
     model Ventdays = triss_prob comorscore2/ clb;
     ods output ParameterEstimates=pe ;
run;
*Results: Beta2 (95% CI) ;

***END OF CODE***;
```

APPENDIX B – NRI/IDI Macro Code

```
/*------------------------------------------------------------------*
   | MACRO NAME  : rocplus
   | SHORT DESC  : Estimate new measures of model improvement
   |               due to Pencina. Integrated Discrimination
   |               Index (IDI)and Net Reclassifcation Improvement
   |               (NRI) for Old model vs New model.
   *------------------------------------------------------------------*
   | CREATED BY  : Bergstralh, Eric            (08/28/2008 14:01)
   *------------------------------------------------------------------*
   | PURPOSE
   |
   | Estimate Integrated Discrimination Index (IDI)and Net
   | Reclassifcation Improvement (NRI) for Old model vs New
   | model. Ref: Pencina, et al. Stat Med. Jan 2008.
   |
   |
   |
   *------------------------------------------------------------------*
   | MODIFIED BY : Kosanke, Jon                (01/15/2009  8:31)
   |
   | Needed to sort __data dataset before running proc logistic.
   *------------------------------------------------------------------*
   | MODIFIED BY : Bergstrahl, Eric            (02/01/2010 13:27)
   |
   | Meet macro standards requirements.
   *------------------------------------------------------------------*
   | OPERATING SYSTEM COMPATIBILITY
   |
   | UNIX SAS v8   :   YES
   | UNIX SAS v9   :   YES
   | MVS SAS v8    :
   | MVS SAS v9    :
   | PC SAS v8     :
   | PC SAS v9     :
   *------------------------------------------------------------------*
   | MACRO CALL
   |
   | %rocplus (
   |            data= ,
   |            event= ,
   |            oldvars= ,
   |            addedvars= ,
   |            idvar= ,
   |            risk= ,
   |            plots=N
   |          );
   *------------------------------------------------------------------*
   | REQUIRED PARAMETERS
```

```
|
| Name       : data
| Default    :
| Type       : Dataset Name
| Purpose    : input data set
|
| Name       : event
| Default    :
| Type       : Variable Name (Single)
| Purpose    : name of numeric event variable, 1=event, 0=no
|               event
|
| Name       : oldvars
| Default    :
| Type       : Variable Name (List)
| Purpose    : list of numeric variables in established (old)
|               logistic model to predict "event" variable
|
| Name       : addedvars
| Default    :
| Type       : Variable Name (List)
| Purpose    : list of numeric variables to be added to the old
|               model. New model predictors = oldvars+addedvars.
|
| Name       : idvar
| Default    :
| Type       : Variable Name (Single)
| Purpose    : name of ID variable, used for merging output
|               data sets from LOGISTIC.
|
*-------------------------------------------------------------*
| OPTIONAL PARAMETERS
|
| Name       : risk
| Default    :
| Type       : Number (Range/List)
| Purpose    : list of cutpoints to categorize event risk
|               probability. Used for NRI estimation.
|
| Name       : plots
| Default    : N
| Type       : Text
| Purpose    : Y or YES is plot of ROC curves, integrated
|               sensitivity and 1-specificity are wanted.
|               Default is N (no plots).
|
*-------------------------------------------------------------*
| RETURNED INFORMATION
|
| Printed summary table with IDI and NRI for old model vs new
|  model with added variables. Classification talbes if risk
|  classes are specified. Plots as requested.
```

```
|
|
*------------------------------------------------------------*
| EXAMPLES
|
| Add clinical gleason score(cl_gleas) to an existing logistic
| regression model to predict failure(slpsa) with 5 years
|  after prostate cancer surgery. Existing risk classes were
|  associated with failure probabilities <.10, .10-.19,
|  .20-.49,.50+.
|
| %rocplus(data=a ,event=slpsa,
|       oldvars= pgs7 pgs810 log2_psa sem_ves mpos
|                ht_adjrtonly,
|       addedvars= cl_gleas,
|       risk= .1 .2 .5,
|       idvar=clinic,
|       plots=Y);
|
|
*------------------------------------------------------------*
| REFERENCES
|
| Ref: Pencina, et al. Stat Med. Jan 2008.
|
|
|
*------------------------------------------------------------*
| Copyright 2010 Mayo Clinic College of Medicine.
|
| This program is free software; you can redistribute it
|  and/or modify it under the terms of the GNU General Public
|  License as published by the Free Software Foundation;
|  either version 2 of the License, or (at your option) any
|  later version.
|
| This program is distributed in the hope that it will be
|  useful, but WITHOUT ANY WARRANTY; without even the implied
|  warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
|  PURPOSE. See the GNU General Public License for more
|  details.
*-----------------------------------------------------------*/

%macro rocplus(data=,event=,oldvars=,addedvars=,risk= ,idvar=,
plots=N);


  *** Input checks ***;
  %local errorflg;
  %let errorflg=0;
  %if &data=  %then %do;
    %put  ERROR - Variable <DATA> not defined;
```

```
 %LET  errorflg = 1;
%end;
%if &event= %then %do;
 %put  ERROR - Variable <EVENT> not defined;
 %LET  errorflg = 1;
%end;
%if &oldvars= %then %do;
 %put  ERROR - Variable <OLDVARS> not defined;
 %LET  errorflg = 1;
%end;
%if &addedvars= %then %do;
 %put  ERROR - Variable <ADDEDVARS> not defined;
 %LET  errorflg = 1;
%end;
%if &idvar= %then %do;
 %put  ERROR - Variable <IDVAR> not defined;
 %LET  errorflg = 1;
%end;

%IF &errorflg=1 %THEN %DO;
   %put ERROR: Macro ROCPLUS not run due to input errors;
   %go to exit;
%end;

%local riske;
%let riske=&risk; %if &riske= %then %let riske=None;

** assign new macro vars as local ***;
 %local ncps aucold aucnew auc_no dfold dfnew dfdiff
     chisqold chisqnew chisqdiff pll n ne
     dnne dne ncne nce upne upe nrie nrine nri nri_se nrie_se
    nrine_se pnri pnrie pnrine ip_o ip_n ip_no ip_pv is_o is_n
    is_no is_pv idi idise idip;

**utility macro to count number words in a string**;
%macro words(string);
 %local count word;
 %let count=1;
 %let word=%qscan(&string,&count,%str( ));
 %do %while(&word ne);
  %let count=%eval(&count+1);
  %let word=%qscan(&string,&count,%str( ));
 %end;
 %eval(&count-1)
%mend words;

%local noldvars naddvars;
%let noldvars=%words(&oldvars);   %put noldvars &noldvars;
%let naddvars=%words(&addedvars); %put naddvars &naddvars;

%if &risk ^= %then %do;
```

```
   %let ncps=%words(&risk); %put ncps &ncps; **no. of risk
categories;
   %do i=1 %to %eval(&ncps-1);  **check risk category ordering**;
    %if %qscan(&risk,&i,' ') >= %qscan(&risk,&i+1,' ') %then %do;
      %Put "WARNING: Risk Category Cutpoints NOT increasing";
    %end;
   %end;
  %end; **end loop for risk groups present;

  **utility macro (cgroup) to group continuous variables;
   %MACRO  CGROUP(V,GROUP,BEGIN,LEN);
     %local be bi i;
     %IF &LEN=  %THEN %LET LEN=16;
     LENGTH &GROUP $ &LEN;
     %LET BE=X;
     %LET I=1;
     %DO %UNTIL (&BE= );
          %LET BI=%qSCAN(&BEGIN,&I,' ');
          %LET BE=%qSCAN(&BEGIN,&I+1,' ');
          %IF &I=1 %THEN %DO;
               IF &V < &BI THEN &GROUP=" <&BI";
          %END;
          %IF &BE^=  %THEN %DO;
               %IF &BE<=&BI %THEN %DO;
               %PUT
           "WARNING: NON-INCREASING INTERVALS IN MACRO _CGROUP
           FOR &GROUP";
               %END;
               IF &BI <=&V< &BE THEN &GROUP="&BI<=X<&BE";
          %END;
          %ELSE %DO;
               IF &V>=&BI THEN &GROUP="&BI+";
          %END;
          %LET I=%EVAL(&I+1);
     %END;
     IF &V=. THEN &GROUP=" ";
   %MEND CGROUP;

data __data; set &data;
  keep &idvar &event &oldvars &addedvars;

  *** delete obs with missing data**;
  if &event not in(0,1) then delete; **1=event, 0=no event;

     **check oldvars for missing;
  if
  %do i=1 %to %eval(&noldvars-1);
     %scan(&oldvars,&i) = . OR
  %end;
     %scan(&oldvars,&noldvars)
   =. then delete;
```

```
      **check added vars for missing;
   if
   %do i=1 %to %eval(&naddvars-1);
      %scan(&addedvars,&i) = . OR
   %end;
      %scan(&addedvars,&naddvars)
   =. then delete;
  run;
  **old model;
  ods listing close;
proc logistic  data=__data descending;
    ods output globaltests=_oldll  association=_cstatold;
   model &event = &oldvars          /outroc=_roco; output
out=_old predicted=pold;
  run;
   ** new model;
proc logistic data=__data  descending;
     ods output globaltests=_newll association=_cstatnew;
   model &event= &oldvars &addedvars /outroc=_rocn; output
out=_new predicted=pnew;
  run;
   ods listing;
   data _oldll;set _oldll;
      if _n_=1;**model chisq log-lik;
      chisqold=chisq; dfold=df;
      drop chisq df;
   run;
   data _newll;set _newll;
      if _n_=1;**model chisq log-lik;
      chisqnew=chisq; dfnew=df;
      drop chisq df;
   run;
   data _cold; set _cstatold;
       if _n_=4;
       cold=cvalue2; keep cold;  **C-stat old model;

   data _cnew; set _cstatnew;
       if _n_=4;
       cnew=cvalue2; keep cnew;  **C-stat new model;
   data _cstat; merge _cold _cnew;
       cdiff=cnew-cold;         **diff in C-stat;
       call symput("aucold",put(cold,5.3));
       call symput("aucnew",put(cnew,5.3));
       call symput("auc_no",put(cdiff,5.3));
   run;
   data _loglik; merge _oldll _newll;
      chisqdif= chisqnew-chisqold; **diff in model chisq;
      dfdif= dfnew-dfold;
      p_ll= 1-probchi(chisqdif,dfdif);
      call symput("dfold", put(dfold,3.));
      call symput("dfnew", put(dfnew,3.));
      call symput("dfdif", put(dfdif,3.));
```

```
      call symput("chisqold", put(chisqold,7.2));
      call symput("chisqnew", put(chisqnew,7.2));
      call symput("chisqdif", put(chisqdif,7.2));
      call symput("pll", put(p_ll,6.4));
   run;

** Basic data for NRI and IDI**;
data _both; merge _old _new; by &idvar;
keep &idvar &event pold pnew n_o sgn;
  N_o=pnew-pold;
**difference in risk for IS and IP calculations;
  sgn=sign(n_o);
   %if &risk= %then %do;
  proc means data=_both noprint  n sum;
        var &event; output out=_both2 n=n sum=ne;
    data _both2; set _both2;
    call symput("n",put(n,7.) );
    call symput("ne",put(ne,7.) );
    run;
   %end;

  %if &risk ^= %then %do;
   ***use apriori categories for risk prob***;
data _class; set _both;
    ***apriori categories of clincal importance**;
    **  Risk categories using old predictor score**;
    if . < pold< %qscan(&risk,1,' ') then old_c=1;
   %do i=1 %to %eval(&ncps-1);
    if %qscan(&risk,&i,' ') le pold lt %qscan(&risk,%eval(&i+1),
     ' ') then old_c=%eval(&i+1);
   %end;
    if pold ge %qscan(&risk,&ncps,' ') then old_c=%eval(&ncps+1);

    ** Risk categories using new predictor score**;
    if . < pnew < %qscan(&risk,1,' ') then new_c=1;
   %do i=1 %to %eval(&ncps-1);
    if %qscan(&risk,&i,' ') le pnew lt %qscan(&risk,%eval(&i+1),
     ' ') then new_c=%eval(&i+1);
   %end;
    if pnew ge %qscan(&risk,&ncps,' ') then new_c=%eval(&ncps+1);

    ** up / down movements by endpoint**;
    if &event=1 then do;
       if new_c gt old_c then do; up_e=1; _v=1; end;
       if new_c lt old_c then do; dn_e=1; _v=-1; end;
       if new_c=old_c    then do; nc_e=1; _v=0; end;
    end;
    if &event=0 then do;
       if new_c gt old_c then do; up_ne=1; _v=1; end;
       if new_c lt old_c then do; dn_ne=1; _v=-1; end;
       if new_c=old_c    then do; nc_ne=1; _v=0; end;
    end;
```

```
   %cgroup(pold,old_cat,&risk);
   %cgroup(pnew,new_cat,&risk);
   run;

**calculate NRI by event & overall**;
 proc means data=_class noprint  n sum;
       var up_e dn_e nc_e up_ne dn_ne nc_ne; output out=_sum
       sum=up_e dn_e nc_e up_ne dn_ne nc_ne;
 data _sum2; set _sum;
     ne= dn_e+nc_e+up_e;    *no events;
     nne=up_ne+nc_ne+dn_ne; *no non-events;
     n= ne + nne;

     nrie= (up_e-dn_e)/ne;
     nrine= (up_ne-dn_ne)/nne ;
     nri= nrie - nrine;                           **equation(8);
     nri_se=   sqrt( (up_e + dn_e) / ne**2 + (up_ne + dn_ne)/
               nne**2 );
     nrie_se=  sqrt( (up_e + dn_e) / ne**2  );
     nrine_se= sqrt( (up_ne + dn_ne)/nne**2 );
     z=     nri /   nri_se;                      **equation (9);
     z_e=  nrie /  nrie_se;                      **equation (10);
     z_ne= nrine / nrine_se;                     **equation (11);
     p_nri=    ( 1- Probnorm(abs(z))    ) * 2;
     p_nrie=   ( 1- Probnorm(abs(z_e)) ) * 2;
     p_nrine= ( 1- Probnorm(abs(z_ne))) * 2;

     call symput("n",put(n,7.) );
     call symput("ne",put(ne,7.) );
     call symput("dnne",put(dn_ne,7.) );
     call symput("dne", put(dn_e,7.) );
     call symput("ncne",put(nc_ne,7.) );
     call symput("nce", put(nc_e,7.) );
     call symput("upne",put(up_ne,7.) );
     call symput("upe", put(up_e,7.) );

     call symput("nrie", put(nrie,7.4) );
     call symput("nrine",put(nrine,7.4) );
     call symput("nri",   put(nri,7.4) );

     call symput("nri_se",  put(nri_se,6.4) );
     call symput("nrie_se",  put(nrie_se,6.4) );
     call symput("nrine_se",  put(nrine_se,6.4) );

     call symput("pnri", put(p_nri,6.4) );
     call symput("pnrie", put(p_nrie,6.4) );
     call symput("pnrine",put(p_nrine,6.4) );
     output;
   * end;
  run; *proc print;
%end; **end loop for risk category present;
```

```
**paired t-test for IS, IP components of IDI;
proc means data=_both noprint n mean std t prt; var pold pnew
n_o; class &event;
 output out=_mnidi n=nold nnew n_n_o  mean=mn_old mn_new mn_n_o
  std=sd_old sd_new sd_n_o
  t=told tnew t_n_o probt=d1 d2 p_n_o;
 * proc print data=_mnidi;
  data _mnidi; set _mnidi;
   if &event=0 then do;
    call symput("IP_O",put(mn_old,7.4) );
    call symput("IP_N",put(mn_new,7.4) );
    call symput("IP_NO",put(mn_n_o,7.4) );
    call symput("IP_pv",put(p_n_o,6.4) );
   end;
   if &event=1 then do;
    call symput("IS_O",put(mn_old,7.4) );
    call symput("IS_N",put(mn_new,7.4) );
    call symput("IS_NO",put(mn_n_o,7.4) );
    call symput("IS_pv",put(p_n_o,6.4) );
   end;
run;

**two sample t for IDI  (15);
ods listing close;
ods output statistics=_stats ttests=_ttest;
proc ttest data=_both;class &event; var n_o;
run;
ods listing ;
 data _stats; set _stats;
   lclass=left(class);
   if substr(lclass,1,4)="Diff" then do;
      IDI=-1*Mean;
      call symput("IDI",put(IDI,7.4) );
      call symput("IDISE", put(Stderr,6.4));
   end;
  data _ttest; set _ttest;
    if method="Pooled";
    call symput("IDIP",put(Probt, 6.4 ));

 *ods pdf file="your path";

data _null_;
 file print;
put / @10 "Beyond the ROC: Incremental Impact of New Model Over
Old Model" /;
put @10 "ROCPLUS macro, Reference: Pencina, et al. Stat Med,
2008:27:157-172." //;
put @10 "Data Set: &data, N=&n";
put @10 "Dependent Variable: &event,  &ne events"/;
put @10 "Apriori event risk category cutpoints(cohort studies):
&riske"/;
put @10 "Old Model variables: &oldvars";
```

```
Put @10 "New Model Variables: &oldvars &addedvars" ///;
put @10 "PERFORMANCE STATISTICS:  OLD vs NEW MODEL "/;
put @10  "--------------- Old Model -------New Model--- New vs.
Old--- P-value ";
put @10
"=================================================================
=====";
put @10 "Model LL Chi-sq  &chisqold        &chisqnew
&chisqdif      &pll";
put @10 "  (DF)             (&dfold)          (&dfnew)
(&dfdif)      ";
put @10 "---------------------------------------------------------
-------------";
put @10  "ROC Curve Area     &AUCold          &AUCnew
&AUC_NO    ";
put @10 "---------------------------------------------------------
-------------";
put @10  "Integ Sens(IS)    &IS_O         &IS_N         &IS_NO
&IS_PV ";
put @10  "Integ 1-Spec(IP)  &IP_O         &IP_N         &IP_NO
&IP_PV ";
put @10  "IDI*              n/a            n/a            &IDI
&IDIP ";
put @10 "   (SE)
(&idise)";
  %if &risk ^= %then %do;
put @10"---------------------------------------------------------
------------";
put @10 "Risk Category Change: Old -->New";
put @10 " Events:";
put @10 "  No. up           n/a            n/a           &upe ";
put @10 "  No. no change    n/a            n/a           &nce ";
put @10 "  No. down         n/a            n/a           &dne ";
put @10 "  NRI-event        n/a            n/a           &nrie
&pnrie ";
put @10 "    (SE)
(&nrie_se) ";
put @10 " Non-Events: ";
put @10 "  No. up           n/a            n/a           &upne ";
put @10 "  No. no change    n/a            n/a           &ncne ";
put @10 "  No. down         n/a            n/a           &dnne ";
put @10 "  NRI-non-event    n/a            n/a           &nrine
&pnrine ";
put @10 "    (SE)
(&nrine_se) ";
put @10 " NRI**                                          &NRI
&pNRI ";
put @10 "   (SE)                                         (&nri_se)
";
   %end;
```

```
put
@10"=============================================================
=========";
put @10  "*IDI=integrated discrimination improvement.  IDI = IS-
IP. Pencina equation 13.";
put @10 " ";
  %if &risk ^= %then %do;
put @10 "**NRI=net reclassification improvement. Pencina equation
8.";
put @10 "  NRI = [P(up|event)-P(down|event) ] - [ P(up|no event)-
P(down|no event) ]";
  %end;
run;
  %if &risk ^= %then %do;
  **track improvements (ups and downs -- (4)----(7);
  proc freq data=_class; tables &event*old_cat*new_cat
                               / norow nocol ;
  run;
 *ods pdf close;
  %end;

 %if %upcase(&plots)=Y or %upcase(&plots)=YES %then %do;
  **PLots**;

***** Make a data set of the current titles *****;
proc sql ;
  create table work._t as select * from dictionary.titles
     where type='T';
  reset noprint;
 quit;
***** Determine # current titles  *****;
proc sql;
   reset noprint;
   select nobs into :T from dictionary.tables
   where libname="WORK" & memname="_T";
 quit;
***** Store titles in macro variables *****;
%LET TITLE1= ;  *Initialize at least one title;
data _null_;
  set _t;
  %IF (&T>=1) %THEN %DO I=1 %TO &T;
     if number=&I then call symput("TITLE&I", trim(left(text)));
     %END;
 run;
***** Macro uses 1 title: see how many current titles can be
retained***;
%LET TNEW = 1;  **#macro titles;
%LET TOTALT = %EVAL(&T + &TNEW);
%IF &TOTALT<=10 %THEN %LET TSHOW=&T;
   %ELSE %LET TSHOW = %EVAL(10 - &TOTALT + &T);
 ***** Step 5) Add your own title(s) to previous titles *****;
%LET NEXTT1=%EVAL(&TSHOW+1);
```

```
** end title fix up **;

data _roc; set _roco(in=inold) _rocn;
  if inold=1 then model="O";
     else model="N";
 symbol v=none i=j c=red;
 symbol2 v=none i=j c=blue l=2;
 axis99 length=4IN;
 axis98 length=6IN;
title&nextt1"ROC Curves for New vs Old(dashed) Models";

proc gplot data=_roc; plot _sensit_*_1mspec_ = model /
vaxis=axis99  haxis=axis99 overlay;
 run;
title&nextt1"Sensitivity vs Predicted Event Prob for New &
Old(dashed) Models";
proc gplot data=_roc; plot _sensit_*_prob_ = model / vaxis=axis99
haxis=axis98 overlay;
run;
title&nextt1" 1-Specificity vs Predicted Event Prob for New &
Old(dashed) Models";
proc gplot data=_roc; plot _1mspec_*_prob_ = model  /
vaxis=axis99 haxis=axis98 overlay;
 run;
***** Restore the previous titles *****;
title1;
%IF (&T>=1) %THEN %DO I=1 %TO &T;
   title&I "&&TITLE&I";
  %END;

 %end; **end of plotting loop;

proc datasets nolist;
      delete _roco _rocn _oldll _newll _cold _cnew _cstat _both
_old _new _both2 _class
          _sum _sum2 _mnidi _stats _ttest _roc
          _cstatnew _cstatold _loglik __data _t;
    run;
%exit:
run;
 %mend rocplus;


***END OF MARCO***;
```