2017

The Intersection Of Chronic Hepatitis C Infection And Cardiovascular Disease

Kimberly Autumn Forde-Mclean
University of Pennsylvania, kimberly.forde@uphs.upenn.edu

The Intersection Of Chronic Hepatitis C Infection And Cardiovascular Disease

Abstract

Hepatitis C virus (HCV) infection is highly prevalent in the US. Though its primary sequelae are liver-related, extrahepatic manifestations contribute to the overall morbidity and mortality of infection. Disorders of lipid metabolism, chronic inflammation and immune dysregulation resulting from chronic infection provide a milieu for extrahepatic manifestations. We sought to examine the role of modification of lipid metabolism on HCV viral load, and to determine the relative contribution of chronic HCV infection to cardiovascular disease.

We first examined the effect of 3-hydroxy-3-methylglutaryl-CoA reductase inhibitors (statins) on HCV viral load. We found that, on average, treatment with at least 30 days of a statin was associated with a lower HCV viral load than that observed in those unexposed to statins. Additionally, while long-term follow up was not available, statin therapy was not associated with an increased incidence of liver injury.

In the second study, we analyzed data from The Health Improvement Network (THIN) to determine if chronic HCV infection was independently associated with incident myocardial infarction (MI). We found no association between chronic HCV infection and incident MI after adjustment for demographics, comorbidities, medication exposures, body mass index (BMI), tobacco use and family history of MI. Additionally, use of a composite cardiovascular endpoint, characterization of medication exposures as time-varying, and accounting for receipt of HCV therapy did not change our findings.

In the conduct of the second study, missing data for BMI were imputed using multiple imputation models. For the third study, we examined whether different variable selection approaches for specification of multiple imputation models result in more or less accurate prediction of investigator-simulated missingness for BMI in THIN. Variable selection procedures utilized to predict missingness included insertion of investigator-chosen variables, use of a high-dimensional approach including all administrative data if a statistical threshold was met, and feature selection driven by machine learning algorithms. We found that the high-dimensional and machine learning approaches, while able to incorporate all data elements, resulted in small improvements in bias but were computationally onerous. The small gains in accuracy achieved with the new methods need to be weighed against the costs of implementation.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Epidemiology & Biostatistics

First Advisor

James D. Lewis

Second Advisor

Sean Hennessy

Keywords

Acute myocardial infarction, Body mass index, Extrahepatic manifestations, Hepatitis C virus, Missing data, Statins

THE INTERSECTION OF CHRONIC HEPATITIS C INFECTION AND CARDIOVASCULAR DISEASE

Kimberly A. Forde-McLean

A DISSERTATION

in

Epidemiology and Biostatistics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2017

Supervisor of Dissertation	Graduate Group Chairperson
_____	_____
James D. Lewis	Nandita Mitra
Professor of Medicine and Epidemiology	Professor of Biostatistics

_____

Dissertation Committee

Sean P. Hennessy, Professor of Epidemiology

David E. Kaplan, Associate Professor of Medicine

Andrea B. Troxel, Professor of Population Health, New York University

THE INTERSECTION OF CHRONIC HEPATITIS C INFECTION AND CARDIOVASCULAR DISEASE

Dedication page

This dissertation is dedicated to my wife and son. Without your love, patience and unwavering support, the completion of this dissertation would not have been possible.

ACKNOWLEDGMENT

I would like to express my sincerest gratitude to Dr. James D. Lewis, my research mentor, professional coach and personal confidant. I thank you for not giving up on me and for reinforcing my faith in my abilities though dark times threatened to end this dissertation journey. Your confidence in me sometimes surpassed my confidence in myself and for that I will be forever grateful.

Thank you to my committee chair, Dr. Sean Hennessy, and my dissertation committee members, Drs. Andrea Troxel and David Kaplan. The hours I spent with you on the various portions of my thesis project were invaluable in making me a better epidemiologist. Additionally, your patience and willingness to meet, even at the final hour, from near or far, is the primary reason for the completion of the work herein. Each of you has been inspirational during this journey and I hope that in this regard I can “pay it forward” to my current and future mentees.

To my clinical mentor, Dr. K. Rajender Reddy, thank you for showing me how to reinvent myself when met with adversity. This thesis and its component parts represent a culmination of many successes in the story of hepatitis C treatment and eventual elimination.

To Michael Harhay and Ravy Vajravelu, I have learned much from your sophisticated yet simple outlook on things large and small. Your knowledge of epidemiology and insightful approach to statistical programming has helped to refine my ability to dissect and perform complex analyses.

To Dr. Vincent Lo Re, my kindred spirit in many ways, you continue to inspire my quest to be a better doctor, better researcher and better person with each of our interactions. It has been my honor to collaborate with you during our many years of working together. I hope that the future will bring more fruitful collaborations.

ABSTRACT

THE INTERSECTION OF CHRONIC HEPATITIS C INFECTION AND CARDIOVASCULAR DISEASE

Kimberly A. Forde-McLean

James D. Lewis

Hepatitis C virus (HCV) infection is highly prevalent in the US. Though its primary sequelae are liver-related, extrahepatic manifestations contribute to the overall morbidity and mortality of infection. Disorders of lipid metabolism, chronic inflammation and immune dysregulation resulting from chronic infection provide a milieu for extrahepatic manifestations. We sought to examine the role of modification of lipid metabolism on HCV viral load, and to determine the relative contribution of chronic HCV infection to cardiovascular disease.

We first examined the effect of 3-hydroxy-3-methylglutaryl-CoA reductase inhibitors (statins) on HCV viral load. We found that, on average, treatment with at least 30 days of a statin was associated with a lower HCV viral load than that observed in those unexposed to statins. Additionally, while long-term follow up was not available, statin therapy was not associated with an increased incidence of liver injury.

In the second study, we analyzed data from The Health Improvement Network (THIN) to determine if chronic HCV infection was independently associated with incident myocardial infarction (MI). We found no association between chronic HCV infection and incident MI after adjustment for demographics, comorbidities, medication exposures, body mass index (BMI), tobacco use and family history of MI. Additionally, use of a

composite cardiovascular endpoint, characterization of medication exposures as time-varying, and accounting for receipt of HCV therapy did not change our findings.

In the conduct of the second study, missing data for BMI were imputed using multiple imputation models. For the third study, we examined whether different variable selection approaches for specification of multiple imputation models result in more or less accurate prediction of investigator-simulated missingness for BMI in THIN. Variable selection procedures utilized to predict missingness included insertion of investigator-chosen variables, use of a high-dimensional approach including all administrative data if a statistical threshold was met, and feature selection driven by machine learning algorithms. We found that the high-dimensional and machine learning approaches, while able to incorporate all data elements, resulted in small improvements in bias but were computationally onerous. The small gains in accuracy achieved with the new methods need to be weighed against the costs of implementation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1: INTRODUCTION

OVERVIEW OF CHRONIC HEPATITIS C VIRUS (HCV) EPIDEMIOLOGY

Hepatitis C virus (HCV) infection is a public health epidemic, associated with an exponentially increasing morbidity and mortality and resulting in approximately 6.5 billion dollars in healthcare expenditures annually.[1-4] The total costs for management of chronic infection were expected to peak at $9.1 billion dollars in year 2014, however, this estimate failed to include the high costs of new direct-acting antiviral (DAA) therapy, which was approved by the Food and Drug Administration (FDA) in the same year.[4,5] Worldwide, approximately 71 million persons are estimated to have been infected with HCV, representing a decline from the 170 million previously projected due to the high mortality associated with infection, aging of the infected population and more accurate estimates of disease incidence and prevalence from regions around the globe.[6] Based on the most updated review of data from the National Health and Nutrition Examination Survey (NHANES), approximately 2.7 million persons in the United States are chronically infected with HCV, making HCV infection one of the most common blood borne infections encountered to date.[7-9]

Because of the unique characteristics of the HCV viral life cycle, including its rapid rate of replication, genetic heterogeneity with 7 major genotypes and 75 subtypes, circulating quasispecies that differ by 1-3%, and successfully employed strategies to evade the host immune response, infection persists chronically in up to 86% of persons who are acutely exposed.[10-17] Once chronic infection is established, the virus infects hepatocytes and a cycle of hepatocyte injury and repair ensues. This process is often silent, with a lack of clinical symptoms being observed in the majority of those persons

with chronic infection. Unfortunately, chronic HCV infection often remains undiagnosed until the complications of cirrhosis and chronic liver disease are manifest. There are data to suggest that just over 50% of people chronically infected with HCV in the US are unaware of their diagnosis.[18] The liver-related complications of HCV include jaundice, ascites, gastrointestinal bleeding and/ or hepatic encephalopathy.[17] Additionally, given the pro-oncogenic milieu that exists in the cirrhotic liver, chronic HCV infection also contributes significantly to the rising incidence of hepatocellular carcinoma being seen in the US and abroad. This manifestation of chronic liver disease contributes to the high rate of mortality in affected patients.[3,19-21]

Mathematical modeling studies have suggested that the proportion of persons with cirrhosis secondary to chronic HCV infection will continue to rise during this decade with the number of cases of cirrhosis and hepatic decompensation peaking in years 2020 and 2022, respectively.[1] However, with the approval of DAA therapy in 2014 by the FDA, greater than 90% of those treated in randomized trials for their chronic HCV infection achieved a sustained virologic response (SVR) or viral cure.[5] Though not yet available to all patients with chronic HCV infection given cost considerations, insurance coverage and concerns about adherence, these newly available DAA regimens not only bring with them the promise of exceptionally high efficacy and safety, but may also temper the significant burden of chronic HCV on those infected and the infection's broader impact on healthcare expenditures.[22] It has been projected that use of DAAs will decrease infection prevalence by 50% and reduce liver-related deaths, and cases of decompensated cirrhosis, hepatocellular carcinoma and need for liver transplantation by 35% by year 2020.[23,24]

CHRONIC HCV INFECTION AND HOST SPECIFIC IMMUNE RESPONSES

Though hepatic manifestations are well characterized in chronic HCV infection, a state of immune up-regulation ensues that contributes to the systemic effects of chronic infection. In acute HCV infection, after viral acquisition occurs and the virus gains access to the blood stream, the virus travels to the liver and begins to infect hepatocytes. HCV replicates continuously once incorporated into the hepatocyte, releasing replicons into the blood stream. Infected hepatocytes, once recognizing the HCV virion as non-self via pattern recognition receptors, like retinoic acid inducible gene-1 (RIG-1), melanoma differentiation-associated protein 5, toll-like receptor 3 and protein kinase PKR, switch on the host's innate immune system machinery by producing interferons (IFN) and more specifically activating interferon stimulating genes (ISGs).[25-27] The resultant up-regulation of gene expression and elaboration of IFN beckons activation of natural killer (NK) cells, the first members of the immune system to defend against this viral pathogen and lyse infected hepatocytes. The NK cells further inhibit viral replication by producing interferon gamma (IFN-γ) and tumor necrosis factor - alpha (TNF-α). Though initially beneficial, NK cells also contribute to the pathogenicity of HCV as they induce inflammation in the liver and modulate hepatic fibrosis by exerting a direct effect on hepatic stellate cells.[28,29]

Antigen-presenting cells such as Kupffer cells and dendritic cells also play a role in the host immunologic response to HCV. Upon activation, these cells may produce inflammatory cytokines such as TNF-α, Interleukin (IL)-10, and IL-12. They also activate CD4+ and CD8+ T cells, a process which is delayed in comparison to the protections offered by the innate immune mechanisms outlined above. Chemokines released from injured hepatocytes serve to recruit additional T cells to lymphoid tissue, where antigen-presenting cells await their arrival. Activated T cells, both CD4+ helper cells and CD8+

cytolytic cells, then circulate back to the liver and mediate further augmentation of the adaptive host immune response. Of note, weak T cell responsiveness may be one of the reasons for failure to clear HCV after acute infection.[30]

HCV skillfully evades the host immune system by employing several distinct yet coordinated mechanisms. Firstly, various regions of the HCV polyprotein effectively inhibit the production of type 1 IFNs. For example, the non-structural (NS) 3/4A protease complex, a target for some of the available DAA therapies, blocks RIG-1 signaling as well as toll-like receptor signaling thereby suppressing not only activation of IFN stimulating genes but also regulation of IFN production.[25] Secondly, HCV can induce up-regulation of IL-10 production, an immunosuppressive cytokine that inhibits production of other cytokines by antigen-presenting cells such as dendritic cells, thus down-regulating signals important in coordinated cellular responses.[27] This process may in turn lead to T cell exhaustion.[27] Thirdly, there have been conflicting reports of an increase in T1 helper responses in some persons with chronic infection while others have an immune response that fails to polarize T1 helper function, a dichotomy which may well be associated with the proclivity of the infection to persist chronically in some persons.[31] Regardless of the mechanism(s) utilized, HCV's ability to alter the immune response contributes to a milieu of immune dysregulation and chronic inflammation, the effects of which not only affect the liver but also directly and indirectly affect many distinct organ systems.

CHRONIC HCV INFECTION AND EXTRAHEPATIC MANIFESTATIONS

Chronic HCV infection has been found to be associated with a number of conditions that can be regarded as extrahepatic manifestations of disease. These manifestations include mixed essential cryoglobulinemia, lymphoid proliferative

disorders, and autoimmune disorders including those affecting the thyroid.[32-34] There is strong evidence that these disorders occur when there is a predominance of T1 helper responses in the HCV infected host. Additionally, these manifestations may also reflect a direct interaction with products of viral replication, interference with cell signaling pathways, a systemic up-regulation of immune function or may represent a reaction to viral replication in end organs.[35]

METABOLIC COMPLICATIONS: INSULIN RESISTANCE/ DIABETES

Multiple studies have demonstrated that chronic HCV infection is a risk factor for metabolic conditions as well. For example, patients with chronic HCV infection are at increased risk for the development of type II diabetes.[36,37] Several lines of evidence suggest that the phenotype of type II diabetes seen in association with chronic HCV infection may not be the same as that seen in the general population. Firstly, there is a correlation between increased expression of TNF-α that occurs with the underlying degree of hepatic fibrosis which is associated with insulin resistance and hence the development of diabetes.[38] Secondly, HCV may exert a direct cytopathic effect on islet cells, a mechanism responsible for a diabetic phenotype more akin to that observed in patients with type I diabetes.[39] Additionally, residence of HCV infected cells in the pancreas can further recruit more immune cells and facilitate immune cell mediated damage and endocrine dysregulation.[40] Lastly, autoantibodies may be produced in the setting of chronic HCV infection and receipt of interferon-based therapy and may therefore be responsible for an autoimmune type injury and result in diabetes mellitus.[41] Regardless of the underlying mechanism, patients with HCV infection are at increased risk of developing diabetes mellitus in their lifetime, with a prevalence of approximately 15%.[34] Furthermore, insulin resistance and diabetes mellitus, when present, are

associated with poor outcomes in chronic HCV infection with an acceleration of hepatic fibrosis and an increase in incidence of HCC noted.[34]

METABOLIC COMPLICATIONS: DYSLIPIDEMIA

Chronic HCV has also been implicated in dysregulation of lipid metabolism and has been observed to cause hepatic steatosis, with these derangements being a direct result of HCV's ability to harness the lipid metabolism machinery to ensure its entry into and out of the hepatocyte, its target cell. Firstly, transport of the HCV virion into the hepatocyte is mediated by several cell surface receptors. As HCV circulates in the serum, it is complexed to very low-density lipoprotein (VLDL) derived particles, thereby creating a lipoprotein complex that can exploit the lipid receptors on the cell surface and permit the virion's entry into the cytosol.[42-44] Additionally, the virus may enter the hepatocyte via a scavenger receptor type B or the Niemann-Pick type C1 like gene receptor, both important in lipid metabolism.[45] Secondly, it has been discovered that apolipoprotein E isoforms are of importance for exit of HCV virions out of the hepatocyte, thereby mediating infectivity of the virus.[46]

Cellular proteins implicated in the replication and assembly of HCV are also involved in normal host cholesterol metabolism. During *in vivo* cholesterol metabolism, adding various chemical subunits alters intracellular proteins. In this process, referred to as prenylation, farnesyl, a 15-carbon subgroup, or geranylgeranyl, a 20- carbon subgroup, is covalently bonded to an intracellular protein. Prenylated proteins, such as protein kinases, subsequently interact with other cellular proteins and facilitate signal transduction and intracellular trafficking.[47] It is the process of geranylgeranylation that has been implicated in HCV viral replication.[48,49] Ye and colleagues were able to not only

disrupt HCV viral replication in cultured hepatoma cells with an inhibitor of cholesterol metabolism but they were able to demonstrate resumption of HCV viral replication with the addition of geranylgeraniol.[50]

Of interest, the use of the host's cholesterol metabolism machinery results in hypolipidemia, with significantly lower levels of total cholesterol, low-density lipoprotein (LDL), high-density lipoprotein (HDL) and triglycerides observed in those with chronic HCV infection, regardless of degree of underlying hepatic fibrosis. However, patients with insulin resistance and HCV tend to have elevated triglycerides, as seen in other patients with insulin resistance.[51] It is unclear whether and how the dyslipidemia resulting from chronic HCV infection affects cardiovascular risk (see below). Regardless, based on the established dependence of HCV replication on beta-lipoproteins, both for cellular entry and exit and viral assembly, the cholesterol metabolism pathway is not only an integral part of disease pathogenesis that results in dyslipidemia and other metabolic complications but is also a logical target for therapeutic intervention, and serves as the basis of the first specific aim presented in this dissertation.

METABOLIC COMPLICATIONS: HEPATIC STEATOSIS

Hepatic steatosis is noted in just over 50% of patients with chronic HCV infection, a prevalence higher than that for other forms of chronic liver disease.[52] Hepatic steatosis seen in chronic HCV infection may occur under 2 settings; 1) infection with genotype 3 virus and 2) steatosis associated with the metabolic syndrome.[52,53] Regardless of the form of hepatic steatosis noted, chronic HCV infection promotes lipogenesis, decreases oxidation of free fatty acids and down-regulates lipid export, predisposing hepatocytes to storage of fat.[53] Presence of the HCV core protein in the hepatocyte inhibits VLDL

secretion and activates peroxisome proliferator-activated receptor (PPAR).[54,55] Though many potential viral mediated pathways may be engaged for the formation of hepatic steatosis, the presence of steatosis increases the risk of progression of hepatic fibrosis and serves as a cardiovascular risk factor.[56]

Given the degree of immune dysregulation and increase in the metabolic conditions in chronic HCV infection, cardiovascular disease (CVD) may be considered an extrahepatic manifestation of chronic HCV infection, either through an increase in cardiovascular risk factors or through an independent effect of chronic infection with HCV. The second specific aim of this dissertation will explore the potential association between chronic HCV infection and CVD in a population-based cohort.

CARDIOVASCULAR DISEASE AS AN EXTRAHEPATIC MANIFESTATION OF CHRONIC HCV INFECTION

Inflammation contributes to the pathophysiology of CVD. The initiation and progression of precursor lesions such as the “fatty streak” are characterized by the recruitment of immune cells, including monocyte derived macrophages and lymphocytes.[57] Given the presence of lymphocytes in the cellular infiltrate noted in atherosclerotic plaques *in vivo*, investigators have hypothesized that infectious agents, specifically bacterial and viral pathogens, may play a role in disease pathogenesis.[58,59]

As outlined above, an integral feature of chronic HCV infection is activation of the immune system, both cellular immunity and elaboration of inflammatory cytokines as characteristic of innate immunity, in response to the viral replication *in vivo.* In addition to augmenting T helper cells, which mediate immune responses in opposition to viral

pathogens, HCV infection is associated with release of a number of pro-inflammatory cytokines.[60] In fact, many of the inflammatory mediators implicated in CVD are also elevated in chronic HCV infection and other chronic immune mediated conditions. Thus, it is plausible that the inflammatory state induced by chronic HCV infection can lead to an increase in the frequency of cardiovascular events.

Several studies have examined the relationship between HCV infection and CVD, as noted at the commencement of this dissertation (**Table 1**).[61-68] Many of these studies demonstrated that HCV infection increased the risk of CVD. Ishizaka et al. conducted a cross sectional study among healthy individuals undergoing routine health screening and found that HCV seropositivity was associated with carotid artery plaque (OR 1.92; 95% CI 1.56 - 2.38) and carotid intima-media thickening (OR 2.85; 95% CI 2.28 - 3.57) after adjusting for known confounders.[64] Another study of similar design conducted by the same group found a strong association between HCV core proteins and carotid plaque (OR 5.61; 95% CI 2.06 - 15.26).[65] A study conducted by Vassalle et al., which included only hospital-based subjects, reported that HCV seropositivity was an independent predictor of coronary artery disease (OR 4.2; 95% CI 1.4 - 13.0).[62]

In contrast, several other studies contradict the aforementioned results. A case-control study conducted by Arcari et al. found no association between HCV seropositivity and acute MI (RR 0.94; 95% CI 0.52 - 1.68) in a cohort of men enlisted in the US military.[63] In addition, a prospective cohort study by Kiechl et al. found no association between chronic hepatitis B virus and/or HCV infection and the development of carotid plaques.[66] However, the study included a mean follow up period of 5 years, which may not be sufficient to observe the cardiovascular effects of chronic HCV infection. Momiyama examined the prevalence of HCV antibody or HBsAg in subjects with

documented CVD and found no association between HCV seropositivity and CVD (OR 1.08; 95% CI 0.3 - 3.7).[61] Lastly, a cohort study conducted by Bilora et al. found that HCV infection was protective against the development of carotid artery plaques; however, this study included a relatively small number of subjects.[68]

Taken together, studies examining the association between HCV infection and CVD as the current work was being undertaken were conflicting and suffered from a number of methodologic limitations, including small sample sizes, a lack of control for confounding variables, and examination of surrogate outcomes (**Table 1.1**). The importance of HCV infection is rooted in its high prevalence in the US and abroad, its association with many extrahepatic conditions, and its increased risk of liver and non-liver related morbidity and mortality, leading to an increase in health care expenditures.[2,34,35] Additionally, in the current era, chronic HCV infection can be cured in >90% of patients prescribed highly effective DAA therapy.[5] Thus, studies that examine the association between chronic HCV infection and extrahepatic complications as well as the likelihood of their regression or cure with eradication of the infection are needed. Furthermore, because CVD is the number 1 cause of death globally, treatment of its risk factors has the potential to saves lives. If chronic HCV infection is a risk factor for CVD, treatment with DAA therapy could reduce both liver and CV-related mortality.

Table 1.1. Studies examining association between HCV infection and cardiovascular disease at the commencement of the thesis

Reference	Design	Sample Size	No. with HCV	Outcome (No.)	OR or RR (95% CI)
Arcari et al.[63] (2006)	Case-control	582	52 total 22 of cases	Myocardial infarction (292)	RR 0.94 (0.5-1.7)
Ishizaka et al.[65] (2003)	Cross sectional	1992	25 total 16 of cases	Carotid plaque (416)	OR 5.61 (2.1-15.3)
Ishizaka et al.[64] (2002)	Cross sectional	4782	104 total 40 of cases	Carotid plaque (1070)	OR 1.92 (1.6-2.4)
Momiyama et al.[61] (2005)	Case-control	630	21 total 18 of cases	Angiographic documentation of CAD (534)	OR 1.08 (0.3-3.7)
Vassalle et al.[62] (2004)	Case-control	614	30 total 26 of cases	Angiographic documentation of CAD (419)	OR 4.2 (1.4-13.0)
Bilora et al.[68] (2002)	Cohort	98	42 total	Carotid artery plaque (38)	RR 0.54 (0.3-0.9)

CONDUCT OF EPIDEMIOLOGIC STUDIES OF CHRONIC HCV INFECTION UTILIZING ADMINISTRATIVE DATA SOURCES

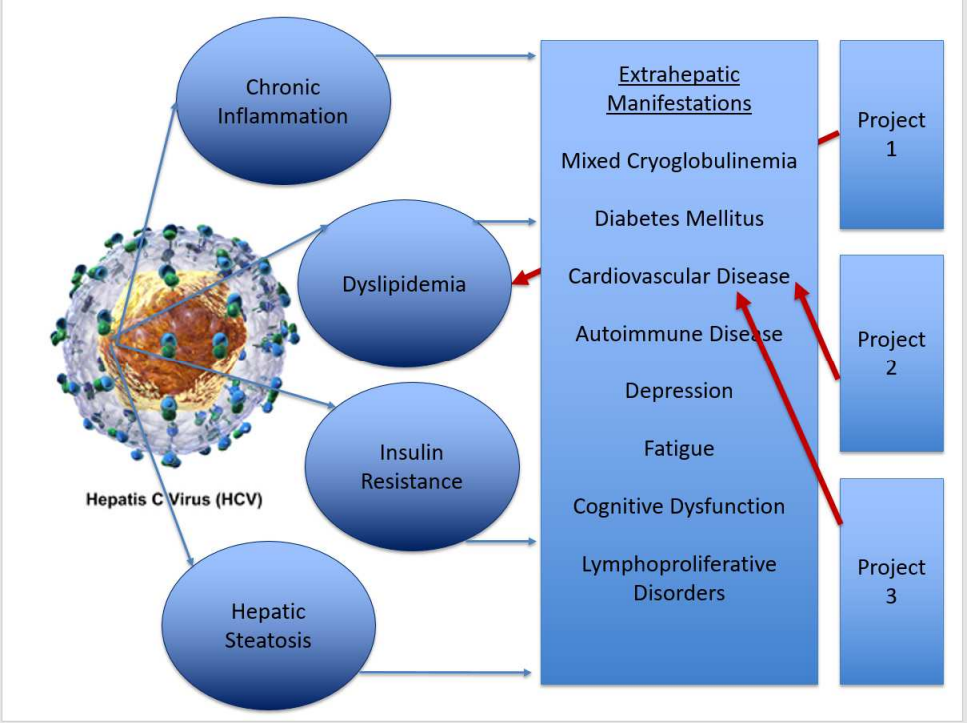
Missing data are a limitation to the study of chronic HCV infection or any chronic medical condition in large electronic medical records databases. Often, there is little information on various aspects of disease pathogenesis including duration of infection, disease stage as based on the assessment of liver biopsy or non-invasive markers of hepatic fibrosis, diagnosis of advanced liver disease including cirrhosis, and diagnosis of hepatic decompensations including ascites, hepatic encephalopathy and gastrointestinal bleeding. Additionally, important confounders for the study of chronic HCV infection, including body mass index (BMI) and smoking are often partially available or not

catalogued as part of the data captured in such databases. Hence, the study of chronic HCV infection epidemiology and its manifestations may be challenging in clinical and claims databases. Statistical techniques for the prediction of important missing confounders in the study of chronic HCV infection will serve as the third and final aim of the herein proposed dissertation.

SIGNIFICANCE OF THE PROPOSED SERIES OF STUDIES

This series of studies will help to characterize the profile of dysregulation of lipid metabolism as seen in a cohort of Veterans with chronic HCV infection, a cohort in whom the prevalence of chronic HCV infection is high and for whom treatment prior to the introduction of DAAs was often delayed as a result of relative and absolute contraindications to therapy.[69-71] The first specific aim of the dissertation will furthermore determine how blockade of the cholesterol machinery, with administration of 3-hydroxy-3-methylgltuarayl coenzyme-A reductase inhibitors or statins, impacts HCV replication as evidenced by HCV RNA or viral load. The second specific aim will examine the question of whether chronic HCV infection is an independent risk factor for myocardial infarction or a composite cardiovascular endpoint after controlling for traditional cardiovascular risk factors in a cohort from the United Kingdom assembled in The Health Improvement Network. The last aim, a methodologic exploration of techniques for variable selection for specification of multiple imputation models, will compare traditional to novel approaches for the prediction of a missing confounder, BMI, a simulation exercise that will be of use in large databases utilized for epidemiologic research but also for continued exploration of chronic HCV infection and its extrahepatic manifestations, a relatively underexplored topic in the current literature (**Figure 1.1**).

Figure 1.1: Conceptual Framework of Thesis Studies



CHAPTER 2: 3-Hydroxy-3-Methylglutaryl Coenzyme A (HMG Co-A) Reductase Inhibitors and their Effect on Hepatitis C RNA in Chronic Hepatitis C Virus (HCV) Infection

Short Title: Statins in Chronic HCV Infection

Authors: Forde, Kimberly A[1,2]; Kaplan, David,[1,3]; Troxel, Andrea B[2,4,5]; Hennessy, Sean P[2,4]; Bewtra, Meenakshi[1,2]; Lewis, James D[1,2]

[1]Division of Gastroenterology, Department of Medicine, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

[2]Center for Clinical Epidemiology and Biostatistics, Department of Biostatistics, Epidemiology and Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

[3]Division of Hepatology, Department of Medicine, Corporal Michael J. Crescenz VA Medical Center, Philadelphia, PA, USA.

[4]Department of Biostatistics, Epidemiology and Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

[5]Division of Biostatistics, Department of Population Health, New York University School of Medicine, New York University, New York, NY, USA.

Abbreviations: Hepatitis C Virus (HCV); Human Immunodeficiency Virus (HIV); International Classification of Disease, 9th Revision Clinical Modification (ICD-9); Low-

density Lipoprotein (LDL); Regional Data Warehouse (RDW); Ribonucleic Acid (RNA); Veterans Affairs (VA)

Corresponding Author: Kimberly A. Forde, 423 Guardian Drive, 722 Blockley Hall, Philadelphia, PA 19104-6021, email: kimberly.forde@uphs.upenn.edu, telephone: 215-746-8597.

Declaration of Conflicts of Interests: The authors have no conflicts of interest to disclose.

Declaration of Funding Interests: This work was supported, in part, by a research grant from the Penn Clinical and Translational Science Award (grant # UL1RR024134 from the National Center For Research Resources), by National Institutes of Health research grant K23-DK090209 (to KAF) and by National Institutes of Health research grant K24-DK078228 (to JDL). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Center For Research Resources or the National Institutes of Health.

Disclaimer: The views expressed in this article are those of the authors and do not necessarily represent the views of the U.S. Department of Veterans Affairs of the U.S. Government.

Abstract:

Background:

Chronic hepatitis C virus (HCV) infection exploits proteins involved in lipid metabolism to support viral replication and assembly. Given preliminary data suggesting that lipid metabolism is altered in the setting of chronic HCV infection and that statins have been demonstrated to interrupt viral replication in cell culture, we undertook a study to determine if, on average, HCV infected patients treated with statins have a lower HCV RNA compared to those untreated.

Methods:

We performed a cross-sectional study using data abstracted from the Veterans Affairs (VA) regional data warehouse (RDW). Adults, ages 18 - 65 years, with chronic HCV infection, as determined by a positive quantitative HCV RNA, from January 2001 through December 2006 were considered. Patients meeting inclusion and exclusion criteria were placed into statin exposure groups (current exposure, former exposure or never exposed) based on presence/ duration of statin exposure and timing of HCV RNA assessment. The primary outcome was average log HCV RNA, as determined by PCR. The effect of statin exposure on HCV RNA was then determined in univariable and multivariable linear regression models, clustered on patient. Sensitivity analyses included assessment of differential drug effects on HCV RNA, and the impact of statin dose and duration.

Results:

Of the 2262 patients meeting the inclusion and exclusion criteria and having appropriate timing of HCV RNA assessment, there were 145 with current statin exposure and 72 with former statin exposure. The statin exposed groups, current and former, were significantly older, had more cardiovascular risk factors and laboratory evidence of dyslipidemia (elevated total cholesterol, LDL and triglycerides). They also had lower liver aminotransferases at baseline. Log HCV RNA was significantly lower with current exposure to statins compared to those never exposed, a finding that persisted after adjustment for age, sex, race, diabetes mellitus and baseline alanine aminotransferase (ALT) level (-0.412, 95% CI -0.681, -0.143). No dose response relationship was found. Therapy for greater than one year in duration was associated with a lower log HCV RNA. Only one patient had a significant increase in liver aminotransferases over the period of statin exposure.

Conclusions:

Exposure to statins was associated with a significantly lower log HCV RNA in comparison to those never exposed. Longer duration of therapy, but not higher dose therapy, was associated with a more significant lowering of log HCV RNA. Elevations in aminotransferases were rare during statin exposure.

Introduction:

Chronic infection with hepatitis C virus (HCV) is a prevalent health condition; with population-based studies suggesting that approximately 1.0-1.6% of the US population is chronically infected.[7-9,72] Unfortunately, approximately 50% of those infected are unaware of their status, precluding them from accessing direct-acting antiviral (DAA) therapy, HCV therapy that is effective at eradicating greater than 90% of chronic infections.[5,18] However, even those who are aware of their infectious status may not have access to DAA therapy if uninsured or therapy fails to be approved by their insurance carriers because of varied insurer practice, concern about adherence in certain demographic groups or failure to meet pre-established standards for medical necessity.[22,73]

Prior to the discovery and subsequent approval of DAA therapy, pegylated interferon and ribavirin served as the standard of care for treatment of chronic HCV infection. Due to limited efficacy and an abundance of potential adverse events, many patients chronically infected with HCV went untreated.[71] During this era, there was much interest in the exploration of alternative therapies, used for off label indications that could provide clinical benefit. Such an interest was garnered in agents that disrupt lipid metabolism.

Basic and translational research suggest that beta-lipoproteins are an integral component of the cellular mechanisms required for *in vivo* HCV viral transport, replication and assembly.[45,74,75] Additionally, the low-density lipoprotein (LDL) receptor is a conduit for endocytosis and transport of the HCV virion across the hepatocyte membrane.[42] Hence, medications that influence the LDL receptor and cholesterol

metabolism more generally may be of significance in the regulation of HCV viral replication. As such, 3-hydroxy-3-methylglutaryl coenzyme-A (HMG Co-A) reductase inhibitors (statins) have garnered much attention.

To extend the observations made in cell culture that HCV RNA replication is disrupted with exposure to statins in infectious clones, and to confirm a dose and agent specific decrease in HCV viral load, we undertook a study to investigate whether, on average, HCV infected patients treated with statins have lower HCV RNA compared to those not treated with this widely used therapy.[50,76]

Materials and Methods:

We performed a cross-sectional study using data abstracted from the Veterans Affairs (VA) regional data warehouse (RDW) maintained in Houston, Texas. VA hospitals around the country provide care to 10,000,000 veterans annually [~170,000 HCV-infected]. Inpatient and outpatient encounters as well as pharmacy data are housed in the centralized database. The VA RDW provides a unique opportunity to examine the effect of statins on HCV viral replication because: 1) there is a high prevalence of chronic HCV infection in veteran cohorts; 2) HCV status can be determined from the laboratory database as well as *International Classification of Diseases, Ninth Revision* (ICD-9) codes; 3) a sufficiently large number of HCV-infected patients receive their medical care at the VA, ensuring an adequate sample size for this study; and 4) clinical variables as well as pharmacy data relevant to study of chronic HCV infection and statin exposure are available for examination.[69,70]

The following patients were eligible for study inclusion: 1) adults aged 18 - 65; 2) chronic HCV infection, as determined by a positive quantitative HCV RNA level, Roche Cobas Amplicor v2.0, determined at least once between January 1, 2001 and December 31, 2006; 3) available pharmacy data; and 4) receipt of care at a veterans integrated services network 4 site inclusive of those VAs serving Pennsylvania, Delaware and regions of Ohio, New Jersey and New York. Patients were excluded if they had comorbid conditions or were exposed to therapies that may have an effect on HCV viral replication or interfere with cholesterol metabolism. Hence, patients were excluded if they had any of the following: 1) human immunodeficiency virus (HIV) as determined by International Classification of Diseases-9th Revision Clinical Modification (ICD-9) codes (042-044); 2) decompensated liver disease, as determined by one inpatient or 2 outpatient ICD-9 code(s) for cirrhosis (571) with or without hepatic decompensation(s) (ascites, hepatic encephalopathy, hepatocellular carcinoma, jaundice, portal hypertension, or variceal hemorrhage; ICD-9 codes 572) or; 3) other chronic liver diseases (alcoholic liver disease, autoimmune disease, hepatitis B, primary biliary cirrhosis, primary sclerosing cholangitis or chronic liver disease otherwise unspecified; or 4) active malignancy noted before the index date, defined as the date of the HCV RNA determination, or within 3 months afterwards.[77] Patients on the following therapeutic agents were also excluded: 1) HCV therapy (i.e. standard interferon monotherapy, standard interferon plus ribavirin, pegylated interferon monotherapy, or pegylated interferon plus ribavirin noted before or on the index date as these constituted the available treatment regimens for chronic HCV infection during the study period); and 2) immunosuppressant medications or conditions for which immunosuppressants are commonly employed including inflammatory bowel disease and organ transplantation.

The primary outcome for the study was a quantitative HCV RNA measurement. Statin exposure categories were established based on use and duration of the exposure in subjects with chronic HCV infection as well as the timing of the assessment of HCV RNA. **Current** exposure was defined as having filled at least 2 prescriptions for statin therapy and having received at least 30 days of cumulative exposure, as validated by pharmacy data, prior to the assessment of the HCV RNA. The HCV RNA measurement however had to be performed while the patient was still in receipt of statin therapy to be considered currently exposed. **Former** exposure was defined as having been prescribed a statin within the 365 days prior to the HCV RNA measurement. For this group, they also required at least 2 consecutive prescriptions for any statin and at least 30 days of cumulative exposure. However, their HCV RNA measurement by definition occurred sometime after the statin discontinuation date but within 365 days. **Never** exposed referred to those subjects who prior to the HCV RNA assessment had never been prescribed statin therapy for any clinical indication. Those patients with a single statin prescription or who were in receipt of non-consecutive statin prescriptions were excluded.

Baseline demographic data including age, sex, race, and ethnicity were collected on all eligible patients. Information on comorbidities such as hypertension, hypercholesterolemia, diabetes mellitus, and coronary artery disease were also collected. Additionally, information on alcohol use disorder and/or remission, as established by ICD-9 codes (291, 303, 305, 425.5, 571.0, 571.1, 571.2, and 571.3) was ascertained. The following laboratory values were collected at or within 365 days of the index date: sodium, creatinine, white blood cell count, hemoglobin, platelet count, alanine aminotransferase (ALT), aspartate aminotransferase (AST), albumin,

prothrombin time (PT), partial thromboplastin time (PTT), international normalized ratio (INR), total cholesterol, low-density lipoprotein (LDL) cholesterol, high-density lipoprotein (HDL) cholesterol, and HCV genotype. All quantitative HCV RNA measurements were obtained throughout the period of observation; when HCV RNA was measured more than once during an exposure window, the first HCV RNA determination was used for data analysis.

Baseline characteristics were compared by statin exposure level, i.e. current exposure, former exposure and never exposed compared overall. Categorical variables were summarized by frequencies and proportions, and continuous variables were summarized by means and standard deviations or medians and interquartile ranges (IQR), depending on the distribution of the variable of interest. Continuous variables were compared using Kruskal Wallis testing. Categorical variables were compared using a chi-square test or Fisher's exact test, as appropriate.

The underlying distribution of the primary outcome variable, HCV RNA, was found to be left skewed, therefore HCV RNA was log transformed for regression analyses. In these analyses, the primary comparison was that of mean log HCV RNA for those subjects with current exposure to statins as compared to those subjects never exposed during the study period. This outcome was also reexamined for those subjects with former exposure in comparison to the never exposed cohort. All comparisons in univariable analysis determined to have a level of significance of less than 0.1 with the Wald test and/or are of biologic importance to the relationship of statin exposure to HCV viral replication were included in multiple linear regression models. Potential confounders identified *a priori* included age, sex, race, HCV genotype, alcohol use/ abuse and ALT level. Any additional confounders identified in univariable analysis were

also included in the model. Additionally, as a few subjects had observations in the current exposure and former exposure groups, all analyses were clustered on patient to account for the inherent correlation between observations. Though the study protocol proposed an analysis limited to those patients with observations in both the current exposure and former exposure groups, there were too few (n=5) observations available to conduct this analysis.

Several sensitivity analyses were performed to further explore the relationship between statin exposure and HCV viral load. Differential effects of various statins preparations, including atorvastatin, lovastatin, and simvastatin, were examined. Dose and duration of therapy were also explored. For the dose analysis, patients were considered to be in receipt of high dose therapy if they were prescribed ≥ 40 mg of the statin in question. While not all statins are equipotent, a dose of ≥ 40 mg daily for the three most commonly used statin preparations at the VA during the study period, atorvastatin, lovastatin, and simvastatin, is considered moderate or high intensity therapy based on the American College of Cardiology/ American Heart Association Guidelines for treatment of elevated cholesterol.[78] Long-term exposure to statins was defined as at least one year of continuous statin therapy and short-term therapy was defined as less than one year but at least 30 days of statin exposure.

Though additional sensitivity analyses were planned to explore ascertainment of pharmacy data, all patients included in the sample had been prescribed at least one additional medication and/ or had a visit in the VA system during the study period. In a secondary analysis, we determined if any patients had an elevated alanine aminotransferase during statin therapy.

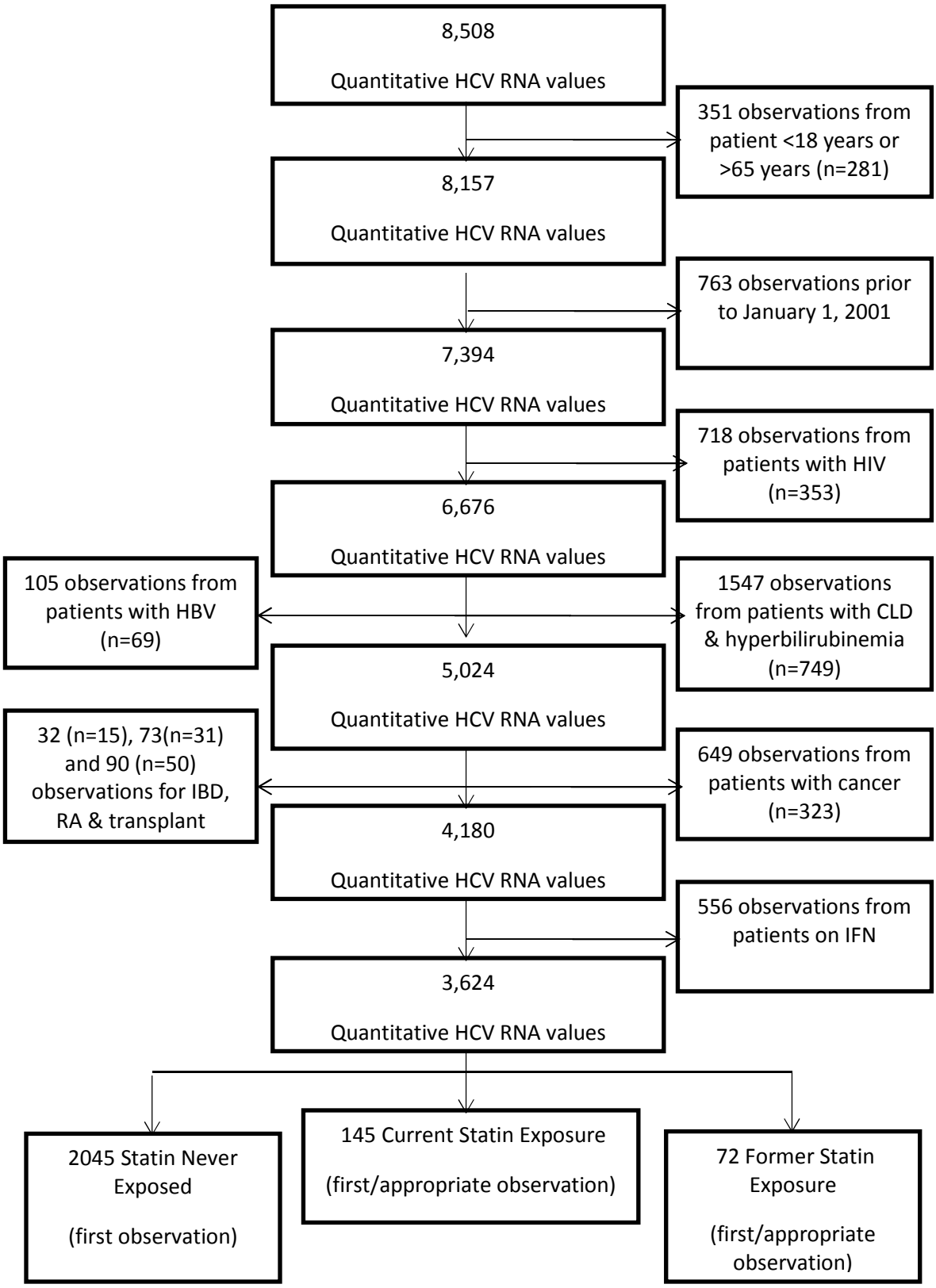
All hypothesis tests were 2-sided and a p-value <0.05 was used to define statistical significance. Analyses were performed using STATA version 13.1 (StataCorp, College Station, TX).

Results:

Of 11,091 patients for whom data were available, there were 8,508 quantitative HCV RNAs results during the study period, of which 3,624 met the inclusion criteria. Reasons for exclusion were as follows: 351 were obtained in patients who were less than 18 years or greater than 65 years of age; 763 were obtained prior to January 1, 2001; the remaining exclusions were from patients with a diagnosis of HIV (718), non-viral chronic liver disease (105); chronic hepatitis B infection (1574); cancer (649); inflammatory bowel disease (32); rheumatoid arthritis (73); an organ transplant (90); and on interferon therapy (556). Limiting to the first HCV RNA assessment, 2045 measures were obtained from those patients who were statin unexposed, 204 from those who were current statin users and 115 from those who were former statin users. After reviewing the length of statin exposure and the timing of the HCV RNA assessment, the current exposure group was reduced to 145 patients (n=33 patients with < 30 days of continuous statin exposure and n=26 with HCV RNA measurement outside of statin exposure window) and the former exposure group to 72 patients (n=10 with cumulative statin exposure of < 30 days and n=33 patients with HCV RNA assessment outside of 365 day window of statin cessation date). Of note, a patient could be represented in multiple statin exposure categories as noted above (n=5). For instance, a single patient may have had an HCV RNA assessment performed while on a statin as well as one after

discontinuation of statin therapy, allowing them to be represented in the current and former statin exposure groups. This strategy was allowed to increase the sample size for each of the statin exposure categories but was also accounted for in adjusted analyses (**Figure 2.1**).

Figure 2.1: Cohort Assembly



The cohort of HCV viremic patients who were never exposed to statins tended to be younger and without significant cardiac risk factors (inclusive of hypertension, hyperlipidemia, diabetes, and prior cardiovascular events) (p<0.001) **Table 2.1**.

Table 2.1. Baseline demographics of cohort and statin exposure groups

Variables	Total Cohort N=2262	Never Exposed N=2045	Current Exposure N=145	Prior Exposure N=72	P-Value
Age *Median (IQR)*	51 (47, 55)	51 (47, 54)	53 (50, 56)	54 (50, 56)	<0.001
Sex *N (%)*					
Male	2176 (96.2)	1962 (95.9)	142 (97.9)	72 (100.0)	0.118
Race *N (%)*					
White	613 (27.1)	549 (26.8)	49 (33.8)	15 (20.8)	0.159
Black	777 (34.3)	700 (34.2)	53 (36.5)	24 (33.3)	
Other	1 (0.1)	1 (0.1)	0 (0.0)	0 (0.0)	
Unknown	871 (38.5)	795 (38.9)	43 (29.7)	33 (45.8)	
Ethnicity *N (%)*					
Hispanic	39 (1.7)	35 (1.7)	2 (1.4)	2 (2.8)	0.613
Co-Morbidities *N (%)*					
Hypertension	945 (41.8)	784 (38.3)	118 (81.4)	43 (59.7)	<0.001
Hyperlipidemia	309 (13.7)	146 (7.1)	148 (74.5)	55 (76.4)	<0.001
Diabetes Mellitus	415 (18.3)	307 (15.0)	76 (52.4)	32 (44.4)	<0.001
Cardiovascular Disease	177 (7.8)	98 (4.8)	57 (39.3)	22 (30.6)	<0.001
Peripheral Vascular Disease	80 (3.5)	59 (2.9)	14 (9.7)	7 (9.7)	<0.001
Obesity	210 (9.3)	166 (8.1)	34 (23.4)	10 (13.9)	<0.001
Renal Disease	55 (2.3)	38 (1.9)	13 (9.0)	2 (2.8)	<0.001
Psychiatric Conditions	1039 (45.9)	923 (45.1)	89 (61.4)	30 (41.7)	0.001
Habits *N (%)*					
Alcohol	1040 (46.0)	956 (46.7)	60 (41.4)	24 (33.3)	0.042
Drugs	1079 (47.7)	991 (48.5)	65 (44.8)	23 (30.9)	0.017
Tobacco	602 (26.6)	534 (26.1)	46 (31.7)	22 (30.6)	0.250
Laboratory Studies*^					
Sodium	138.6 (3.0)	138.6 (2.9)	138.3 (3.2)	138.5 (4.0)	0.004
Creatinine	1.0 (0.9, 1,1)	1.0 (0.9, 1.1)	1.0 (0.9, 1.2)	1.0 (0.9, 1.1)	<0.001
Total Bilirubin	0.8 (0.6, 1.0)	0.8 (0.6, 1.0)	0.7 (0.6, 0.9)	0.8 (0.7, 1.0)	0.032
Albumin	4.0 (3.8, 4.3)	4.0 (3.8, 4.3)	3.9 (3.6, 4.1)	4.1 (3.9, 4.3)	<0.001
ALT	49 (33, 76)	50 (33, 78)	35 (23, 56)	45 (29, 67)	<0.001
AST	41 (30, 63)	42 (31, 65)	30 (24, 43)	38 (27, 55)	<0.001
INR	1.0 (1.0, 1.1)	1.0 (0.9, 1.1)	1.0 (1.0, 1.1)	1.0 (1.0, 1.2)	0.011
Platelets	226 (186, 270)	226 (186, 268)	227 (196, 283)	241 (190, 265)	0.195
Total Cholesterol	174 (37)	173 (35)	183 (48)	181 (47)	0.062
LDL	103 (34)	102 (32)	107 (43)	118 (37)	0.002
HDL	42 (34, 51)	42 (34, 52)	36 (32, 45)	36 (31, 43)	0.001
Triglycerides	108 (76, 157)	105 (74, 153)	132 (97, 203)	120 (87, 183)	0.001
HCV RNA					
Log IU/mL	5.9 (5.5, 6.6)	5.9 (5,5. 6.6)	6.0 (4.1, 6.8)	6.1 (5.3, 6.8)	0.195
HCV Genotype					
1	859 (38.0)	790 (38.6)	42 (29.0)	27 (37.5)	0.914
2	97 (4.3)	89 (4.4)	5 (3.4)	3 (4.2)	
3	48 (2.1)	45 (2.2)	3 (2.1)	0 (0.0)	
4	8 (0.3)	8 (0.4)	0 (0.0)	0 (0.0)	
Unknown	1250 (55.3)	1113 (54.4)	95 (65.5)	42 (58.3)	

*Values reported as means (standard deviation) or median (interquartile range) depending on distribution

^Laboratory abbreviations: Aspartate aminotransferase (AST), alanine aminotransferase (ALT), international normalized ratio (INR), low-density lipoprotein (LDL), high-density lipoprotein (HDL)

However, patients never exposed to a statin were more likely to carry diagnoses of alcohol and drug use. Of the statin exposure groups, those currently on statin therapy were more likely to have cardiac comorbidities and elevated measures of LDL, total cholesterol and triglycerides.

The median log HCV RNA was 5.9 IU/mL (IQR 5.5, 6.6) in never exposed patients, 6.0 IU/mL (IQR 4.1, 6.8) in those with current statin exposure and 6.1 IU/mL (IQR 5.3, 6.8) in those with former statin exposure, a non-significant finding when all groups were compared. On univariable analysis, in which current exposure was compared to never exposed and former exposure was compared to never exposed, current exposure to statins was associated with a 0.323 lower log HCV RNA (95% CI -0.588, -0.059). Former exposure was not associated with a lower log HCV RNA. On univariable analysis, other significant variables associated with log HCV RNA included older age, female sex, and black and other race ($p<0.001$ for all comparisons). There were slightly higher log HCV RNA values noted in those with hypertension (0.148, 95% CI 0.048, 0.248), and diabetes mellitus (0.143, 95% CI -0.021, 0.265). Expectedly, elevated AST (0.003, 95% CI 0.002, 0.004) and ALT (0.001, 95% CI 0.001, 0.003) values were associated with a higher log HCV RNA. In multivariable analysis, adjusting for age, sex, race, diabetes and ALT at baseline, and clustering on patient, current statin exposure was significantly associated with lower log HCV RNA (-0.412 log unit lower compared to those never exposed, 95% CI -0.681, -0.143). In contrast, former statin exposure was not associated with log HCV RNA levels (-0.206, 95% CI -0.537, 0.124), **Table 2.2**.

Table 2.2: Univariable. Multivariable Linear Regression: Effect of Statins on HCV RNA

Variable	Univariable Analysis			Multivariable Analysis*		
	Log HCV RNA	95% CI	P-Value	Log HCV RNA	95% CI	P-Value
Age	0.028	0.018, 0.038	<0.001	0.018	-0.008, -0.027	0.020
Sex						
Female	-1.129	-1.488, -0.774	<0.001	-0. 753	-1.080, -0.427	<0.001
Race						
Black	0.419	0.288, 0.551	<0.001	0.370	0.238, 0.503	<0.001
Other	1.469	1.360, 1.578	<0.001	1.230	1.083, 1.377	<0.001
Unknown	0.405	0.271, 0.538	<0.001	0.369	0.237, 0.488	<0.001
Ethnicity						
Hispanic	0.233	-0.024, 0.490	0.075			
Comorbidities						
Hypertension	0.148	0.048, 0.248	0.004			
Hyperlipidemia	-0.130	-0.213, 0.033	0.117			
Diabetes Mellitus	0.143	-0.021, 0.265	0.022			
Cardiovascular Disease	0.085	-0.099, 0.270	0.362			
Peripheral Vascular Disease	-0.075	-0.375, 0.221	0.662			
Obesity	0.067	-0.114, 0.249	0.467			
Renal Disease	0.052	-0.233, 0.338	0.718			
Psychiatric Conditions	0.015	-0.085, 0.116	0.764			
Habits						
Alcohol	0.010	-0.080, 0.110	0.839			
Drugs	0.071	-0.028, 0.170	0.162			
Laboratory Studies^						
AST	0.003	0.002, 0.004	<0.001			
ALT	0.001	0.001, 0.003	0.075	-0.086	-0.273, 0.100	0.363
Total Cholesterol	-0.001	-0.003, 0.005	0.199			
LDL	-0.001	-0.002, 0.001	0.437			
HDL	0.002	-0.002, 0.006	0.324			
Statin Exposure						
Current	-0.323	-0.588, -0.059	0.017	-0.412	-0.681, -0.143	0.003
Former	-0.029	-0.570, 0.289	0.858	-0.206	-0.537, 0.124	0.221

^ Laboratory abbreviations: Aspartate aminotransferase (AST); Alanine aminotransferase (ALT); Low-density lipoprotein (LDL); High-density lipoprotein (HDL)
*Adjusted for age, sex, race, diabetes and ALT at baseline

In sensitivity analysis, the effect of various statin preparations was evaluated. Based on the availability of statin preparations on the VA formulary during the study period, only atorvastatin, lovastatin and simvastatin could be assessed. In univariable analysis, simvastatin was associated with a statistically significant lower log HCV RNA

(-0.292, 95% CI -0.508, -0.077). In adjusted analyses accounting for age, sex, race, diabetes, and ALT, the association between simvastatin and log HCV RNA remained statistically significant (-0.381 log unit lower log HCV RNA, 95% CI -0.663, -0.098). The sample sizes in the groups exposed to lovastatin (n=17) and atorvastatin (n=3) were too small to glean any useful information about administration of these drugs and their effect on log HCV RNA (**Table 2.3**).

There was no clear dose-response relationship demonstrated when comparing statin exposed patients taking 40 mg or more daily to those never exposed (**Table 2.4**). Receiving more than one year of therapy however resulted in a significantly lower log HCV RNA in univariable and multivariable analysis when compared to no therapy. Less than one year of therapy was not associated with log HCV RNA levels (**Table 2.5**).

Table 2.3. Statin Specific HCV RNA Lowering Effect

	Univariable Analysis			Multivariable Analysis*		
Drug	**Log HCV RNA**	**95% CI**	**P-Value**	**Log HCV RNA**	**95% CI**	**P-Value**
Current Statin Use						
Atorvastatin (n= 3)	-0.329	-1.709, 0.992	0.603	-0.481	-2.052, 1.090	0.548
Lovastatin (n= 17)	-0.545	-1.114, 0.024	0.061	-0.666	-1.526, -0.195	0.129
Simvastatin (n=125)	-0.292	-0.508, -0.077	0.008	-0.381	-0.663, -0.098	0.008
No Statin Use	Reference			Reference		

*Adjusted for age, sex, race, diabetes and ALT at baseline

Table 2.4: Dose Specific Statin HCV Lowering Effect

	Univariable Analysis			Multivariable Analysis*		
Dose	**Log HCV RNA**	**95% CI**	**P-Value**	**Log HCV RNA**	**95% CI**	**P-Value**
Current Use						
High Dose^ (n=64)	-0.205	-0.501, 0.092	0.176	-0.265	-0.556, 0.025	0.074
Low Dose (n=81)	-0.417	-0.682, -0.152	0.002	-0.542	-0.807, -0.276	<0.000
No Statin Use	Reference			Reference		

^High dose was considered any of the statin preparations prescribed at a dose of ≥ 40 mg daily
*Adjusted for age, sex, race, diabetes and ALT at baseline

Table 2.5. Effect of Statin Duration on HCV RNA Lowering Effect

Statin Duration	Univariable Analysis			Multivariable Analysis*		
	Log HCV RNA	95% CI	P-Value	Log HCV RNA	95% CI	P-Value
Current Use						
1 year or greater (n=49)	-0.329	-0.556, -0.102	0.004	-0.403	-0.630, -0.176	0.001
Less than 1 year (n=96)	-0.303	-0.713, 0.107	0.147	-0.264	-0.867, -0.062	0.024
No Statin Use	Reference			Reference		

*Adjusted for age, sex, race, diabetes and ALT at baseline

In review of liver aminotransferase levels, only one patient was observed to have had an elevation in the ALT while on statin therapy. Therapy was discontinued on the same date as the ALT laboratory date. Though the aminotransferases normalized following this observation, clinical information about the patient's course was not available for review.

Discussion:

In this cross-sectional study of patients chronically infected with HCV, exposure to statins for at least 30 days in duration was associated with a 0.412 lower log HCV RNA when compared to those never exposed over the study period. While we had small numbers of patients on statin preparations other than simvastatin, our data suggests that the effect is likely to be one of class rather than an agent-specific effect on HCV RNA. Additionally, a clear dose-response relationship was not demonstrated when high dose and low dose therapy were compared to no statin use, suggesting that even low dose therapy has an effect on HCV viral load. Lastly, we demonstrated that a statin duration of greater than one year was associated with a significantly lower log HCV RNA,

advocating for prolonged and continuous use of statins to derive the potential HCV lowering effect afforded.

The biological relationship between cholesterol metabolism and HCV life cycle has been well established. The HCV virion can form a complex with LDL and has been observed to circulate in the sera in this conformation.[74,75] Additionally, such complexes can facilitate passage of HCV across the hepatocyte membrane via the LDL receptor and the type B HDL scavenger receptor, providing a direct route for transit of infectious material into the hepatocyte, its favored target.[42,45] Furthermore, the cellular machinery involved in cholesterol metabolism has also been implicated in HCV viral replication. Prenylation of intracellular proteins is a process by which carbon subgroups are bonded covalently to intracellular proteins, thereby affecting cellular trafficking and cell signaling. Farnesyl and geranylgeranyl are two such carbon subgroups that may be added to intracellular proteins. Geranylgeranyl has been implicated in HCV replication *in vitro* and furthermore, use of inhibitors of cholesterol metabolism have been shown to disrupt HCV replication at this step.[48,50]

Statins may also play a role in the modulation of host immune responses and the development of disease complications, particularly hepatic decompensation and death in patients with chronic liver disease from HCV. In addition to altering the process of prenylation, statins disrupt cholesterol rich membrane rafts that assemble within cells. This, as well as the prenylation pathway, not only have effects on HCV viral replication but also modulate host immune responses. In addition to the modulation of inflammatory cytokines, these processes are also essential for the regulation of cells implicated in innate and adaptive immunity including macrophages and CD4+ T cells. These pleotropic effects of statins have been explored in malignancy, autoimmune diseases as

well as infectious processes such as sepsis with equivocal results.[79-83] Lastly, statins exhibit direct effects on the kinetics of chronic liver disease, with a decrease in portal hypertension noted in patients on therapy.[84,85] A recent publication reported that statin therapy was associated with a 40% reduction in hepatic decompensation and death in a cohort of patients chronically infected with HCV.[86]

Though our study failed to demonstrate a differential effect on HCV RNA with different statin preparations, *in vitro* data suggests that the greater the hepatic metabolism required for a particular statin, the more potent the agent may be for modulating HCV viral replication.[76] Rosuvastatin, the most potent statin for alteration in HCV replication kinetics, and pravastatin, the least potent, could not be explored in this study given that they are were not on VA formulary at the time of study. While this as well as the small number of patients on some agents, i.e. atorvastatin and lovastatin, was a limitation in this study, high statin doses were provided to patients with only one patient having an elevated aminotransferase level in the setting therapy. This adds to the small body of literature that suggests that statins are of little harm in the context of the potential benefit afforded in the setting of chronic liver disease.[87]

The study had a number of other limitations. The first limitation is that of potential confounding. Though we carefully obtained data on potential confounders such as alcohol misuse, a factor associated with HCV viral replication as well as likelihood of being in receipt of statin therapy though it may be clinically indicated, there is likely to be residual confounding.[88,89] In the case of alcohol, an exposure that was determined based on the presence of pre-identified ICD-9 codes, there is poor correlation with the diagnosis, as validated by survey data, and the appearance of diagnostic codes in the electronic medical record.[90] Unfortunately, we did not have other means to determine

alcohol misuse as we did not have direct access to patient charts nor did we have information on the Alcohol Use Disorders Identification Test (AUDIT-C) which is now commonplace in VA data.[91] However, regardless of this shortcoming, we would not expect that the effect of alcohol would have been of such a large magnitude that it would have changed the results obtained. Additionally, with the data available for misuse, alcohol did not appear to be a confounder in our data. The second concern is that of selection bias. It is a concern that the statin exposed and unexposed groups are not in fact comparable and do not arise from the same underlying source population. For example, there may be concern about prescribing a statin in a patient with elevated transaminases and/ or decompensated liver disease given concern about drug-induced liver injury. In the conduct of the study, we excluded patients with decompensated liver disease. Additionally, though we did not match on aminotransferase levels or exclude those patients with elevated aminotransferases because as many as 80% of patients with HCV viremia will have an elevation in their ALT, the exposure groups were noted to have comparable distributions of aminotransferases. Lastly, there may have been misclassification of statin exposure if patients in the never or former exposed statin groups were obtaining their statin prescriptions outside of the VA system. However, most VA patients fill their prescriptions at the VA given their reduced cost, and the findings in the study would have been biased towards the null if patients in the other exposure groups were also in receipt of statin therapy.[92]

The study, while it has many limitations, also has a number of strengths. Firstly, given the comprehensive nature of the VA central database, laboratory data were readily available for analysis. As such, our sample size was quite large for the primary analyses, albeit with lower power for secondary and sensitivity analyses. For those patients who

received statins through the VA, accurate prescription data was available for the construction of the exposure windows, crucial for this analysis. Additionally, many studies have been conducted in the VA database and a majority of the ICD-9 codes used for the determination of important comorbidities/ confounders have been validated in VA data.[77]

Lastly, it is important to highlight the fact that this was an observational study and not a randomized controlled trial, the gold standard for the assessment of interventions and therapies. Employing such a study design would have allowed for assessment of HCV RNA prior to, while on, and after cessation of statin therapy as well as determine if a dose response relationship is plausible. Furthermore, safety in this patient population could have been assessed. However, in the setting of concern about the untoward effects of statins in patients with chronic liver disease, and time and monetary constraints surrounding conduct of this study, the observational design employed was a logical first step.

In summary, statin therapy for at least 30 days in duration was cross-sectionally associated with a lower log HCV RNA in patients with chronic HCV infection. There was no association with former vs. never exposure to statins and log HCV RNA. While there was no clear agent-specific effect or dose-response relationship demonstrated, longer duration of therapy, specifically greater than 1 year, resulted in a larger difference in log HCV RNA. These data help to support a hypothesis of a direct effect of statin therapy on HCV viral replication, an association previously described *in vitro* and in cell culture, and an effect that has subsequently been applied clinically in an effort to improve rates of cure with antiviral therapy.[93-96]

CHAPTER 3: Risk of Myocardial Infarction Associated with Chronic Hepatitis C Virus Infection: A Population-Based Cohort Study

Short Title: Hepatitis C and Myocardial Infarction

Authors: Forde, Kimberly A[1,2,3]; Haynes, Kevin[2,3]; Troxel, Andrea B[2,3,4]; Trooskin, Stacey[5] Osterman, Mark[1,2,3]; Kimmel, Stephen[,2,3,6]; Lewis, James D[1,2,3]; Lo Re III, Vincent[2,3,5]

[1]Division of Gastroenterology, Department of Medicine, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

[2]Center for Clinical Epidemiology and Biostatistics, Department of Biostatistics, Epidemiology and Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

[3]Department of Biostatistics, Epidemiology and Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

[4]Division of Biostatistics, Department of Population Health, New York University School of Medicine, New York University, New York, NY, USA.

[5] Division of Infectious Diseases, Department of Medicine, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

[6] Division of Cardiovascular Medicine, Department of Medicine, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

Published in Journal of Viral Hepatitis 2012; 19(4):271- 277.

Keywords: Hepatitis C virus; myocardial infarction; inflammation

Abbreviations: Hazard Ratios (HRs); General Practitioner (GP); Hepatitis C Virus (HCV); Myocardial Infarction (MI); The Health Improvement Network (THIN); United Kingdom (UK)

Corresponding Author: Kimberly A. Forde, 423 Guardian Drive, 722 Blockley Hall, Philadelphia, PA 19104-6021, email: kimberly.forde@uphs.upenn.edu, telephone: 215-746-8597.

Acknowledgement: The study was funded by grants from the Penn Clinical and Translational Science Award, Penn Center for Education and Research on Therapeutics, National Institute of Allergy and Infectious Diseases, and National Institute of Diabetes and Digestive and Kidney Diseases.

Declaration of Conflicts of Interest: The authors have no conflicts of interest to disclose.

Declaration of Funding Interests: This work was supported, in part, by a research grant from the Penn Clinical and Translational Science Award (grant # UL1RR024134 from the National Center For Research Resources), by an Agency for Healthcare Research and Quality (AHRQ) Centers for Education and Research on Therapeutics cooperative agreement (grant #HS10399), by National Institutes of Health research grant K24-DK078228 (to JDL), and by National Institutes of Health research grant K01-AI070001 (to VLR). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Center for Research Resources or the National Institutes of Health.

Abstract:

Background:

Hepatitis C virus (HCV) infection is associated with systemic inflammation and metabolic complications that might predispose patients to atherosclerosis. However, it remains unclear if HCV infection increases the risk of acute myocardial infarction (MI).

Methods:

To determine whether HCV infection is an independent risk factor for acute MI among adults followed in general practices in the United Kingdom (UK), a retrospective cohort study was conducted in The Health Improvement Network (THIN), from 1996 through 2008. Patients ≥18 years of age with at least 6 months of follow-up and without a prior history of MI were eligible for study inclusion. HCV-infected individuals, identified with previously validated HCV diagnostic codes (n=4,809), were matched on age, sex, and practice with up to 15 randomly selected patients without HCV (n=71,668). Rates of incident MI among patients with and without a diagnosis of HCV infection were calculated. Adjusted hazard ratios (HRs) were estimated using Cox proportional hazards regression, controlling for established cardiovascular risk factors.

Results:

During a median follow-up of 3.2 years, there was no difference in the incidence rates of MI between HCV-infected and uninfected patients (1.02 versus 0.92 events per 1,000 person-years; p=0.7). HCV infection was not associated with an increased risk of incident MI (adjusted HR, 1.10; 95% confidence interval (CI), 0.67 to 1.83). Sensitivity

analyses including the exploration of a composite outcome of acute MI and coronary interventions yielded similar results (adjusted HR, 1.16; 95% CI, 0.77 to 1.74).

Conclusions:

In conclusion, HCV infection was not associated with an increased risk of incident MI.

Introduction:

Hepatitis C virus (HCV) infection affects up to 1.6% of the adult population in the United States and 1% in the United Kingdom (UK).[8,97] After exposure, the majority of HCV-infected patients develop chronic infection, manifested by the persistence of HCV RNA in the blood.[9,17,98-101] HCV exerts its main effects on the liver, inducing inflammation that leads to progressive hepatic fibrosis and ultimately cirrhosis in approximately 20% of those chronically infected.[17] HCV infection may also affect organ systems outside of the liver and induce direct or indirect effects on dermatologic, endocrine, hematologic, neurologic, renal, and ophthalmic function.[102] However, its impact on cardiovascular disease remains unclear.

A number of factors related to chronic HCV infection have been hypothesized to contribute to atherosclerosis. HCV infection stimulates the host immune response, activating T helper cells and releasing a number of pro-inflammatory cytokines, including interferon-alpha, interleukin-1, interleukin-6, and tumor necrosis factor-alpha.[60] Since inflammation is important to the development of atherosclerosis and ultimately myocardial infarction (MI), [103-105] the inflammatory state associated with HCV infection might contribute to an increased cardiovascular disease risk. Furthermore, HCV infection has been associated with metabolic complications, including diabetes mellitus,[36,37,106] the metabolic syndrome,[107] and hepatic steatosis,[108] all of which are important risk factors for the development of cardiovascular and peripheral vascular disease.

Existing studies examining the association between HCV infection and cardiovascular diseases have reported conflicting results,[61-68,109-111] and the impact of HCV infection on acute MI has been evaluated primarily among men.[63,111] Given the

prevalence of HCV infection, affecting approximately 170 million people worldwide,[98] and the morbidity and mortality associated with cardiovascular disease, it is important to determine if HCV infection increases the risk of MI among HCV-infected individuals. Thus, our primary objective was to examine whether HCV infection is an independent risk factor for MI within a broadly representative population-based cohort.

Materials and Methods:

Data Source

The Health Improvement Network (THIN) is a database of electronic medical records on over 7.5 million patients from over 1,500 general practitioners (GPs) in 415 UK practices.[112],[113] Data recorded in THIN include demographic information, medical diagnoses, lifestyle characteristics, measurements taken during medical practice, prescriptions, laboratory results, and coded free text comments. Diagnoses are recorded using the Read diagnostic code scheme,[114] and prescriptions are recorded using codes from the UK Prescription Pricing Authority.[115] This study was approved by the University of Pennsylvania Institutional Review Board.

Study Design and Subjects

We conducted a matched retrospective cohort study among patients in THIN aged 18 years or older who were registered with a THIN practice for at least 6 months. Patients were identified as HCV-infected if they had: 1) a diagnosis of HCV infection, or 2) a diagnosis of nonspecific viral hepatitis with “hepatitis C” noted in a free text comment field.[116] HCV-uninfected patients had no diagnosis recorded for either HCV

infection or another cause of viral hepatitis during follow-up. Patients were excluded if prior to the start of follow-up (defined below), they had a diagnosis of: 1) MI, or 2) active hepatitis B virus infection, defined by diagnostic codes for chronic hepatitis B or a positive hepatitis B surface antigen.

All eligible HCV-infected patients were selected and matched to randomly selected HCV-uninfected patients based on age (+/- 5 years), sex, and THIN practice.[117] Up to fifteen HCV-uninfected patients were matched to each HCV-infected subject to ensure sufficient sample sizes for planned subanalyses.

Follow-up for HCV-infected patients began on the date of initial HCV diagnosis or the registration date plus 180 days, whichever was later. Follow-up for HCV-uninfected patients began on the same date as that of their matched HCV-infected subject. Follow-up continued until an acute MI, death, transfer out of THIN, end of study data (November 5, 2008), or end of data collection for the THIN practice. Subjects whose principal cause of death was an acute MI were classified as having this endpoint on their death date.

Main Outcome Measures

The primary outcome was first occurrence of an acute MI after the start of follow-up. Patients were classified as having an acute MI if they received a Read code consistent with this diagnosis during follow-up.[118]

As a secondary outcome measure, we examined a composite outcome of either incident MI or revascularization procedure (e.g., percutaneous transluminal coronary angioplasty, coronary artery bypass grafting) during follow up.

Measurement of Covariates

We collected the following data on or prior to the start of follow-up: age; sex; height; weight; family history of cardiovascular disease; diagnoses of diabetes, hypertension, hyperlipidemia, and chronic kidney disease, defined by diagnostic codes or prescriptions for relevant medications; tobacco, cocaine, and alcohol use, as assessed by the general practitioner; and selected prescriptions (aspirin, non-aspirin non-steroidal anti-inflammatory agents, 3-hydroxy-3-methyl-glutaryl [HMG]-CoA reductase inhibitors, oral hypoglycemic agents, insulin, anti-hypertensive drugs). Patients were considered exposed to a medication of interest at the start of follow-up if a prescription was recorded within 90 days prior to the start of follow-up. All prescriptions for aspirin were collected during follow-up to determine continued exposure.

Statistical Analyses

Incidence rates[119] of MI with 95% confidence intervals (CIs) were determined for HCV-infected and -uninfected subjects. Hazard ratios (HRs) with 95% CIs of first incident MI and a composite outcome of first incident MI or coronary revascularization procedure were estimated using Cox proportional hazards regression adjusted for the matching variables, so that standard errors appropriately reflected the clustering induced by the matched sets.[120] HRs were adjusted for established cardiovascular risk factors (i.e., hypertension, diabetes, hyperlipidemia, family history of cardiovascular disease, and smoking). Additional potential confounding variables evaluated included: age; sex; body mass index (BMI); alcohol consumption; cocaine use; chronic kidney disease; and use of a medication of interest. Confounders were retained in the model if their inclusion changed the unadjusted HR of incident acute MI by more than 15% or were proposed *a priori.*[121] We also assessed interactions between HCV infection and both age and sex. Standard model checking procedures were employed, including visual inspection of

diagnostic log-log plots. Missing values of height and weight were multiply imputed based on fully observed covariates including age and sex.[122,123] The imputation algorithm employed the Markov chain Monte Carlo method and 20 imputed data sets were created.[124,125] Final estimates were obtained using standard formulae to combine estimates from the 20 analyses.[122] All reported results derive from the imputed data sets.

We performed several sensitivity analyses to determine the robustness of our results. We repeated analyses treating aspirin as a time-varying covariate. We performed a sub-analysis evaluating the risk of incident MI among patients documented as having chronic HCV infection by their GP compared to uninfected persons. Finally, since antiviral therapy for HCV infection might affect the risk of acute MI, we repeated our analyses excluding patients who received standard or pegylated interferon prior to or during follow-up.

Assuming an incidence rate of acute MI of 1.33 per 1,000 person-years[126], an average follow-up of 2.5 person-years, and a 15:1 ratio of unexposed to exposed subjects, we estimated that 3,000 HCV-infected patients would provide 80% power to detect a relative hazard of acute MI of 2.0 between HCV-infected and -uninfected patients, using a two-sided, 0.05-level test. Analyses were performed using Stata version 11.0 (StataCorp, College Station, TX).

Results:

Among 4.5 million patients with at least 6 months of follow-up in THIN between February 1996 and November 2008, 5,218 HCV-infected individuals were identified. A total of 40 patients were excluded due to an acute MI recorded prior to the start of follow-up or prior to their HCV diagnosis, 31 were excluded because a date of HCV diagnosis

was not available, 214 for active hepatitis B virus infection, and 124 for an age below 18 years. Hence, 4,809 HCV-infected subjects were matched to 71,668 HCV-uninfected patients.

The characteristics of the HCV-infected and -uninfected subjects are shown in **Table 3.1**. Compared to HCV-uninfected individuals, patients with HCV infection more frequently had diabetes mellitus and chronic kidney disease but less often had hyperlipidemia. HCV-infected patients also more commonly had a lower BMI, were smokers, drank alcohol, used cocaine, and received prescriptions for aspirin and an antihypertensive medication compared to HCV-uninfected patients.

Table 3.1. Baseline characteristics of the HCV-infected and -uninfected cohorts.

Characteristic	HCV-Infected (n=4,809)	HCV-Uninfected (n=71,668)	P-Value
Median age (years, IQR)	38.60 (31.57, 46.68)	38.57 (31.39, 46.45)	0.9
Sex (no., %)			
Male	2,935 (61.03)	43,802 (61.12)	0.9
Body mass index category (no., %)			
Underweight (<18.5 kg/m^2)	172 (3.58)	1,624 (2.27)	<0.001
Ideal (18.5 kg/m^2 – 24.9 kg/m^2)	2,061 (42.86)	26,384 (36.81)	
Overweight (25.0 kg/m^2 – 29.9 kg/m^2)	1,031 (21.44)	17,982 (25.09)	
Obese (>30.0 kg/m^2)	557 (11.58)	9,914 (13.83)	
Missing	988 (20.54)	15,764 (22.00)	
Family history of cardiovascular disease (no., %)	607 (12.62)	9,210 (12.82)	0.7
Medical comorbidity (no., %)			
Diabetes mellitus	259 (5.39)	2,310 (3.22)	<0.001
Hypertension	466 (9.69)	7,168 (10.00)	0.50
Hyperlipidemia	574 (11.94)	9,421 (13.15)	0.02
Chronic kidney disease	99 (2.06)	529 (0.74)	<0.001
Medication use (no., %)			
Aspirin	118 (2.65)	1,303 (1.82)	0.002
Anti-hypertensives	489 (10.17)	5,088 (7.10)	<0.001
HMG-CoA reductase inhibitors	90 (1.87)	1,907 (2.66)	0.001
Hypoglycemic agents	151 (3.14)	1,228 (1.71)	<0.001
Smoking (no., %)			
Ever	3,574 (74.32)	44,423 (61.98)	<0.001
Alcohol consumption (no., %)	3,870 (80.47)	53,200 (74.23)	<0.001
Cocaine use (no., %)	79 (1.64)	16 (0.02)	<0.001
Follow-up time (years, IQR)	2.41 (0.93, 5.10)	3.22 (1.34, 5.83)	<0.001

IQR=interquartile range; HMG=3-hydroxy-3-methyl-glutaryl

During a median follow-up of 2.41 years for HCV-infected and 3.22 years for HCV-uninfected patients, 264 subjects had an incident acute MI (16 HCV-infected versus 248 HCV-uninfected; p=0.9). The incidence rate of acute MI was not statistically different between HCV-infected and -uninfected persons (1.02 versus 0.92 events per 1,000 person-years; p=0.67).

Results examining the association between HCV infection and acute MI are summarized in **Table 3.2**. In unadjusted analysis, HCV infection was not associated with an increase in the risk of incident MI (HR, 1.12; 95% CI 0.68 to 1.84). After controlling for established cardiovascular risk factors including age, sex, hypertension, diabetes, hyperlipidemia, family history of cardiovascular disease, and smoking as well as chronic kidney disease, BMI, and baseline aspirin use, the only additional confounding variables identified, HCV infection was not associated with an increase in the risk of acute MI (adjusted HR, 1.10; 95% CI 0.67 - 1.83). Similar results were observed when examining the association between HCV infection and a composite outcome of first acute MI or a revascularization procedure (adjusted HR, 1.16; 95% CI 0.77 - 1.74). Furthermore, stratification of the results based on age category (less than 50, 50-65, and greater than 65 years) and sex did not change the results (data not shown).

Sensitivity analyses including aspirin as a time-varying covariate did not appreciably alter the results (adjusted HR, 1.07; 95% CI 0.64 - 1.78). After exclusion of subjects who received antiviral therapy for HCV infection prior to or during follow-up, the overall results remained unchanged (adjusted HR 1.13; 95% CI 0.68 - 1.87). Sub-analyses examining the risk of acute MI between patients documented as having chronic HCV by their GP compared to uninfected persons showed similar results to the primary analysis (adjusted HR 0.67; 95% CI 0.16 - 2.71). Finally, given that HCV infection may

increase acute MI risk through diabetes or decrease this risk through hypolipidemia and may therefore be in the causal pathway, we re-ran multivariable models without adjusting for diabetes or hyperlipidemia. No appreciable change in the risk of acute MI was observed (adjusted HR 1.10; 95% CI 0.67 - 1.83 and adjusted HR 1.10; 95% CI 0.66 - 1.82 for diabetes and hyperlipidemia, respectively).

Table 3.2: Unadjusted and adjusted hazard ratios of the risk of first incident myocardial infarction for baseline variables of interest.

Variable	Unadjusted Hazard Ratio*	95% CI	Adjusted Hazard Ratio†	95% CI
Hepatitis C virus infection	1.12	0.68-1.84	1.10	0.67-1.83
Sex				
Male	Reference		Reference	
Female	0.45	0.33-0.62	0.35	0.26-0.47
Age				
<50 years	Reference		Reference	
50-65 years	4.08	3.05-5.45	3.37	2.48-4.57
>65 years	9.68	7.10-13.18	9.25	6.48-13.19
Medical comorbidities				
Hypertension	3.17	2.40-4.19	1.04	0.73-1.49
Diabetes mellitus	2.69	1.66-4.37	0.97	0.57-1.64
Hyperlipidemia	3.18	2.38-4.24	1.35	0.96-1.90
Chronic kidney disease	7.10	3.84-13.11	3.67	1.96-6.83
Aspirin use	4.82	3.00-7.76	0.99	0.57-1.72
Body mass index (BMI) category				
18.5-24.9 kg/m^2	Reference		Reference	
25.0-29.9 kg/m^2	1.93	1.43-2.60	1.47	1.08-1.99
> 30.0 kg/m^2	2.06	1.48-2.87	1.59	1.13-2.24
Ever smoking	1.69	1.28-2.24	1.33	1.00-1.78
Family history of cardiovascular disease	2.46	1.87-3.24	2.22	1.66-2.99

*Body mass index imputed based on height, weight and potential covariates for all subjects for whom a BMI was not available in the dataset

†Adjusted model includes age, sex, co-morbidities as defined by medication usage 90 days prior to the index date or a diagnostic code entered into the medical record prior to the start of follow-up, aspirin use in the 90 days prior to the index date, body mass index measured closest to baseline or imputed if missing and tobacco prior to the start of follow-up

Discussion:

A number of chronic inflammatory diseases, including psoriasis, rheumatoid arthritis, and systemic lupus erythematosus, have been associated with an increased risk of MI.[127-132] However, in this retrospective analysis of HCV-infected and HCV-uninfected patients followed in UK general practices, HCV infection was not associated with an increased incidence of either acute MI or a composite outcome of MI and coronary revascularization procedures. This suggests that not all chronic inflammatory conditions are associated with cardiovascular disease. The hypothesized association between HCV infection and MI was not observed despite the increased prevalence of several known cardiovascular risk factors among the HCV-infected patients, including diabetes, hypertension, and smoking.

Despite the hypothesized link between HCV infection and atherosclerosis, our results suggest that HCV infection does not increase the incidence of acute MI. Although HCV infection stimulates an inflammatory cascade, the resulting inflammation may not be of the magnitude, severity, or subtype sufficient to accelerate atherosclerosis and increase the risk of cardiovascular events. Further, cytokine receptor function and intracellular signaling may not be equally distorted in HCV infection as it is in other chronic inflammatory conditions. In addition, the lower serum lipid levels among the HCV-infected persons, which might be due to binding of HCV to low-density lipoprotein-C receptors or impairment of hepatic assembly of very low-density lipoproteins [133], may counteract any pro-atherosclerotic effects of HCV-associated inflammation. These factors might explain the lack of association between HCV infection and acute MI in this study.

Our results are consistent with those of Arcari et al.,[63] who found no association between HCV infection and acute MI in a case-control study of 582 males in the US

Army. However, our finding that HCV infection was not associated with an increase in the risk of incident MI disagrees with those of several other studies.[62,64,106] Vassalle et al. reported that HCV infection was an independent risk factor for angiographically-documented coronary artery disease in a case-control study of 686 patients (adjusted odds ratio, 4.2; 95% CI 1.4 - 13.0).[62] Ishizaka et al. reported an association between HCV infection and carotid artery plaque and thickening of the intima media.[64] Finally, Butt et al. examined over 170,000 U.S. veterans over a 5-year period and observed that HCV infection was associated with a 27% increase in the incidence of cardiovascular events, defined as myocardial infarction, congestive heart failure, or coronary artery bypass grafting or percutaneous transluminal coronary angioplasty.[111] Our results might differ from these studies because of differences in the populations examined, outcomes evaluated, and confounding variables included in these analyses.

The current study has a number of strengths. It included data from over 80,000 patients followed in general practices in the United Kingdom. Since THIN's general practitioners are provided with incentives to maintain the electronic medical record, information within THIN is recorded with a high degree of accuracy and GPs will have recorded information in the same manner between HCV-infected and -uninfected patients.[112] Finally, our analyses controlled for a number of established cardiovascular risk factors and other important confounding variables that might influence the incidence of MI, including baseline as well as chronic exposure to aspirin, and our results were robust to multiple sensitivity analyses.

There are several potential limitations to this study. First, it is possible that some patients who spontaneously cleared HCV infection were included within the exposed group. However, 54 - 86% of patients infected with HCV develop chronic infection.[17]

Further, we examined the association between documented chronic HCV infection and incident acute MI and demonstrated similar results to those of our primary analyses. In addition, no HCV-uninfected patient was identified as being HCV antibody or RNA positive, minimizing the likelihood of misclassification of HCV status. Second, the duration of follow-up for both cohorts was short, and it remains unclear how HCV infection might affect the risk of MI over a longer duration of time. However, stratifying our results by age categories did not yield statistically significant differences and one can expect that the older population has had a longer duration of infection. Third, unmeasured confounding by unmeasured or unassessed confounders is always possible in observational studies. However, such confounding would not only have to be of considerable magnitude but also be substantially independent of the comprehensive list of factors already included to unmask an association between HCV infection and incident MI. Finally, height and/or weight results were missing in 20% of patients, but multiple imputation was performed to ensure that missing data did not affect the validity of our results.

In conclusion, HCV infection was not associated with an increase in the risk of incident acute MI among a large sample of patients followed in UK general practices. The reduced lipid levels observed among HCV-infected persons might be sufficient to mitigate any pro-atherosclerotic effects of HCV-associated inflammation. The inflammation stimulated by HCV infection may also be different from that of other chronic inflammatory diseases. Regardless of the reason for the lack of association, these data suggest that not all chronic inflammation is associated with an increased risk of cardiovascular disease.

CHAPTER 4: Prediction of Body Mass Index (BMI) in The Health Improvement Network (THIN): Novel Approaches to Variable Selection for Multiple Imputation Models

Short Title: Variable Selection for Prediction of BMI in THIN

Authors: Forde, Kimberly A[1,2,3]; Vajravelu, Ravy K[1,2,3]; Harhay, Michael[2,3]; Lewis, James D[1,2,3]; Hennessy, Sean P[2,3]; Troxel, Andrea B[2,3,4]

[1]Division of Gastroenterology, Department of Medicine, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

[2]Center for Clinical Epidemiology and Biostatistics, Department of Biostatistics, Epidemiology and Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

[3]Department of Biostatistics, Epidemiology and Informatics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, USA.

[4]Division of Biostatistics, Department of Population Health, New York University School of Medicine, New York University, New York, NY, USA.

Keywords: Body mass index; the health improvement network; multiple imputation; variable selection

Abbreviations: Body Mass Index (BMI); Machine Learning (ML); The Health Improvement Network (THIN); United Kingdom (UK)

Corresponding Author: Kimberly A. Forde, 423 Guardian Drive, 722 Blockley Hall, Philadelphia, PA 19104-6021, email: kimberly.forde@uphs.upenn.edu, telephone: 215-746-8597.

Acknowledgement: This study was funded by grants from the Penn Clinical and Translational Science Award, Penn Center for Education and Research on Therapeutics, and National Institute of Diabetes and Digestive and Kidney Diseases.

Declaration of Conflicts of Interest: The authors have no conflicts of interest to disclose.

Declaration of Funding Interests: This work was supported, in part, by a research grant from the Penn Clinical and Translational Science Award (grant # UL1RR024134 from the National Center for Research Resources), by National Institutes of Health research grant K23-DK090209 (to KAF), National Institutes of Health training 5T32DK007066-42 (to RKV) and by National Institutes of Health research grant K24-DK078228 (to JDL). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Center for Research Resources or the National Institutes of Health.

Abstract:

Background:

An elevation in body mass index (BMI), a simple and readily derived index of weight to height, is associated with adverse health outcomes including metabolic diseases, such as diabetes mellitus, cardiovascular disease and cancer. BMI is therefore an important potential confounder in the epidemiologic examination of these disease processes. However, data on BMI are often missing, creating a concern about the generation of biased estimates of association.

Methods:

Utilizing data from The Health Improvement Network (THIN), we examined 3 separate approaches for the selection of variables for multiple imputation models for the prediction of BMI in a continuous or categorical fashion. The approaches included: 1) a traditional investigator- specified approach; 2) a high dimensional selection approach derived from the hierarchical structure of the database and based upon an *a priori* established statistical threshold; and 3) variable selection determined by machine learning algorithms. We conducted a simulation study in which BMI, now well ascertained in THIN since being prioritized as a quality indictor in 2008, was modeled as missing under the three standard assumptions of missingness, including missing completely at random, missing at random and missing not at random. The performance of each approach for the selection of variables for multiple imputation model specification was compared by calculation of estimated percent bias and standardized mean squared error for continuous prediction and percent correctly classified for dichotomous prediction.

Results:

A dataset inclusive of complete and plausible values of BMI for 203,622 patients served as the parent dataset for all simulations. Random samples of 20,000 patients were extracted for each of the 1000 simulations performed for each mechanism of missingness and for each pattern further specified within the mechanism of missingness. Estimates of percent bias were small when considering the high dimensional approach and machine learning approaches, though the generation of estimates of percent bias, standardized mean squared error and percent correctly predicted were computationally onerous.

Conclusions:

The use of alternative approaches to variable selection for multiple imputation model specification afforded less biased and more precise estimates. However, the improvement in prediction and minimization of bias need to weighed against the computational costs of the applied alternative methods.

Introduction:

The Quetelet Index, renamed the body mass index (BMI) in 1972, is a simple ratio of weight in kilograms to height squared in meters.[134,135] Since the observation that being overweight, defined as a BMI ≥ 25.0 kg/m^2, or obese, defined as a BMI ≥ 30.0 kg/m^2, are associated with adverse health outcomes, BMI has frequently been utilized in medicine, the social sciences and other industries such as insurance, for assessment of risk and determination of likelihood of premature death. BMI is an important risk factor for metabolic disorders and cardiovascular diseases, hence its availability for use is of utmost importance in the epidemiologic study of these chronic, non-communicable conditions. Of note, BMI was an important potential confounder in the relationship between chronic HCV infection and incident acute myocardial infarction, the second aim of this thesis and presented herein, and how its missingness was handled, as just over 20% of BMI values were missing, presented a methodologic issue which the investigators had to overcome.

The handling of missing data is an unavoidable challenge in observational studies, particularly those that depend largely upon the use of electronic health records. Though allowing for the conduct of studies with large sample sizes and therefore affording the ability to detect small effect sizes, such large data sources are neither organized nor managed for the collection of clinical variables to be used in the conduct of epidemiologic research. When data on important confounders are incomplete, the investigator is compelled to identify methods to control for or obviate the missingness of that exposure, outcome or confounder. If not addressed, the measurement of an association of interest between an exposure and outcome may be biased.

Many statistical techniques may be applied for handling of missing data, including use of the complete case analysis, creation of a missing category indicator, or performance of single imputation; however, none of these techniques is more desirable than having a complete and accurate dataset.[136] Through the advent of more advanced statistical techniques in the last 3 decades, strategies such as multiple imputation now represent a standard and less biased approach for handing missing data.[122]

While frequently employed in epidemiologic research, there are relatively few guidelines regarding variable selection for the specification of multiple imputation models. Additionally, generation of predictive models to impute missing data are dependent on the availability of many covariates, including data on exposure and outcome of the topic being studied. Furthermore, the very assessment of the accuracy and fit of these models is impossible if these key data are not present for review.

Given these knowledge gaps, the objectives of the proposed study were to evaluate the predictive ability of multiple imputation models utilizing differing approaches to variable selection, inclusive of data elements such as Read diagnosis codes, procedural codes, and prescriptions, to impute missing values for BMI in The Heath Improvement Network (THIN), a United Kingdom (UK) electronic medical record database. THIN, though inclusive of several years of missing data on BMI, instituted collection of this covariate as a quality measure in year 2008, making assessment of the “true” or gold standard values of BMI feasible. We explored different variable selection methods for the imputation of BMI including a traditional investigator-specified approach, high dimensional selection approach in which all data elements were explored but included only if a statistical threshold was reached and a machine learning approach in 1000 random samples of THIN data in which mechanisms of missingness were simulated.

Materials and Methods:

Data Source

The Health Improvement Network (THIN) is a United Kingdom (UK) database that encompasses anonymized data for the conduct of epidemiologic studies worldwide. The database includes data derived from the electronic medical records of over 7.5 million patients from over 1,500 general practitioners (GPs) in the UK and includes over 10 million patient years of data.[112,113] At any one time, more than 3 million subjects are actively being followed in the database. Recorded data elements include demographic information, medical diagnoses, lifestyle characteristics, anthropometric measurements obtained during clinical practice, prescriptions, laboratory results, and free text comments. Diagnoses are recorded using the Read diagnostic code scheme,[114] and prescriptions are recorded using codes from the UK Prescription Pricing Authority.[115] This study was approved by the University of Pennsylvania Institutional Review Board under exempt status. The study was additionally approved by the Scientific Review Committee as administered by IMS Health Real World Evidence Solutions.

Study Population

We identified patients who were active in THIN with a registration date on or after January 1, 2005 and before December 31, 2008. They also had to have an end of follow up date that was greater than December 31, 2008 (end date, transfer date or death date greater than 12/31/2008). These restrictions were placed on the creation of the study population to ensure that all included patients would have been followed before and after prioritization of BMI as a quality indicator in the database and hence increase the chance that data were available from patients who were likely to have an assessment of BMI.

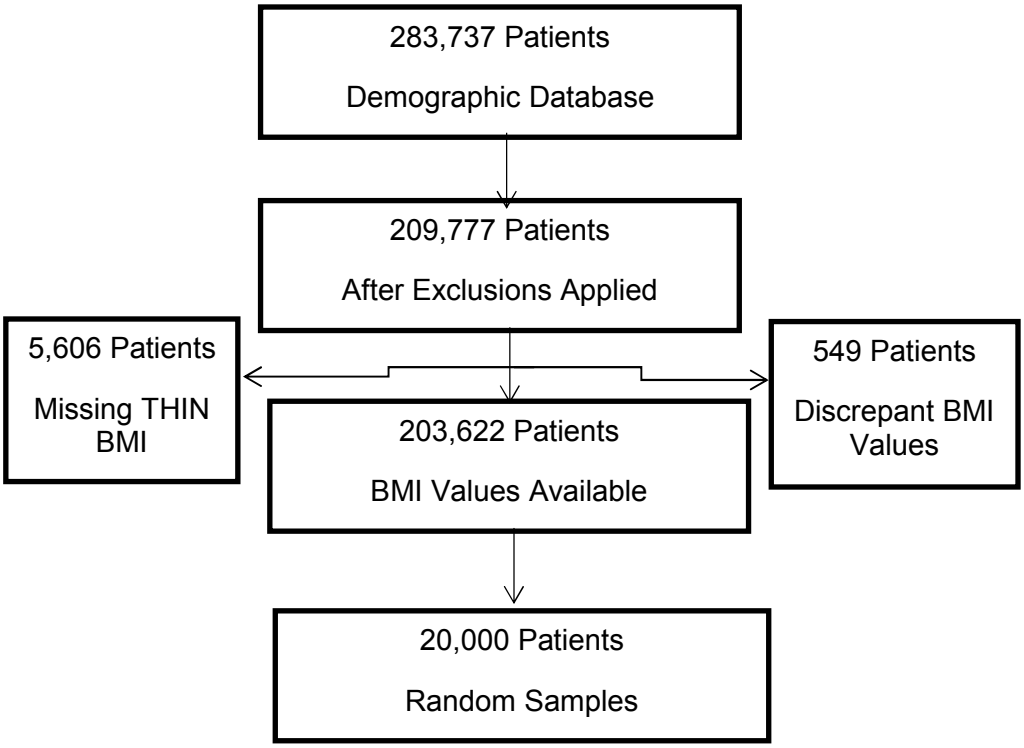
The end of this time frame also corresponded with the end of follow up in the second aim of this dissertation.

After assembling the cohort as specified above, a random sample of 10%, of which the sampling was subdivided by practice to ensure representation of patients from all practices in the final dataset, was obtained. The sample was selected using a random number generator. For all of the THIN patients identified, the demographic, medication diagnosis, therapy, consult, and additional health data (AHD) files were obtained.

Creation of Analytic Datasets

Based on the search strategy outlined above, a master dataset inclusive of 283,737 patients was obtained. After excluding observations from those patients <18 and > 90 years of age (n=11,321); missing BMI provided by THIN and not calculable due to absence of height or weight data (n=52,527); implausible BMI values (including those observations with height or weight of zero, height <1.2 meters, a calculated BMI < 10 or a calculated BMI > 65, n=10,112); missing BMI as provided by THIN, suggesting that the provided data on height/ weight may be erroneous (n=5,606); or discrepant BMI values (BMI calculated from the provided height and weight data failed to match the BMI field provided in THIN, n=549), 203,622 patients remained. For the conduct of the missingness mechanism simulations, random samples of 20,000 patients were drawn for each simulation (**Figure 4.1**).

Figure 4.1: Cohort Assembly

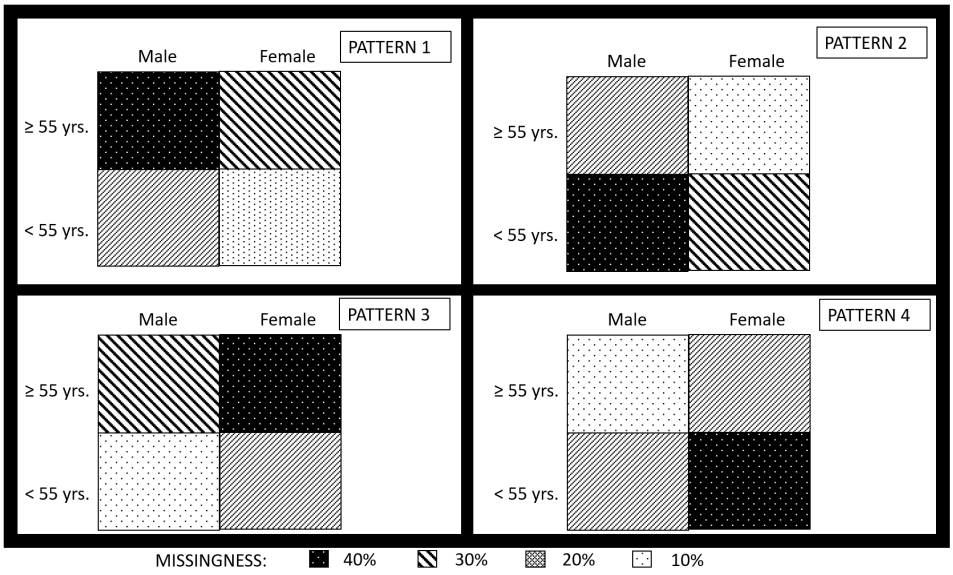


Simulated Mechanisms of Missingness

Mechanisms of missingness were modeled as described by Rubin and Little.[123] Based on "Inference and missing data,[137] three distinct mechanisms of missingness were simulated: 1) missing completely at random (MCAR) in which the data represented a simple random sample of the complete data [**f(R | X, Y)= f(R)** where Y is the outcome, R is the missingness indicator variable (1=observed; 0=missing), X represents the fixed covariates and f(R) is a random sample of the observed data]; 2) missing at random (MAR) in which the probability of missingness is dependent on observed covariates or outcomes [[**f(R | X, Y)= f(R| X, Y_{obs})**]; and 3) Missing not at random (MNAR) in which the probability of a missing observation is dependent upon unobserved variables [**f(R |**

X, Y) = f(R | X, Y_{obs} , Y_{mis})]. Operationally, the distribution of missing BMI values was modeled for MCAR by utilizing a random number generator and deleting BMI values with random assignment to a number of 0.20 or less, thereby creating approximately 20% missingness, the degree of missingness noted in the BMI variable during the conduct of the chronic HCV and incident acute MI study presented in this dissertation. MAR was modeled by choosing age, dichotomized at its median (55 years), and sex as the observed variables upon which missingness were based. Under MAR, 4 patterns of missingness were modeled which reflected more or less missingness based on combinations of age and sex (**Figure 4.2**).

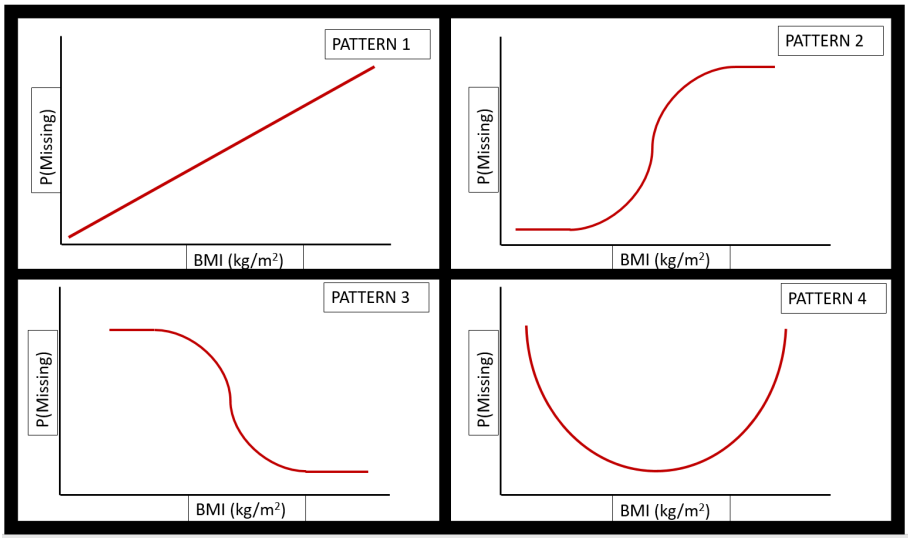
Figure 4.2: Patterns 1-4 of missingness modeled under the MAR mechanism, as determined by combinations of age, dichotomized at 55 years, and sex



For MNAR, missing BMI values were modeled as missing based on the absolute value of BMI. Again, as was performed for MAR, 4 patterns of missingness under this mechanism were simulated. The first included a linear relationship between increasing BMI and missingness. The second and third patterns used a threshold for assigning

missing values (pattern 2 included a threshold BMI of 30 kg/m^2 after which the percentage of missingness was increased and pattern 3 included a threshold of lowest BMI to a cut point of 30 kg/m^2, a window in which missingness was maximized). The last pattern of missingness reflected a bimodal distribution of missing values in which the extremes of BMI were more likely to be missing (**Figure 4.3**).

Figure 4.3: Patterns 1-4 of missingness modeled under the MNAR mechanism



Strategies for Variable Selection for Multiple Imputations Models

Traditional Strategy

In multiple imputation, the approach to variable selection for specification of the imputation model has been relatively unexplored. While some investigators utilize logic-based algorithms for inclusion of variables in their imputation models, the most conservative and unbiased method is to include all available variables in the model.[122] While perhaps prone to over-fitting, this strategy at least ensures that there will not be omission of important variables in the imputation model.[138] The limitation, however, is that in large datasets such as those utilized in epidemiologic research, the inclusion of all

potential variables may not be feasible. In such cases, it has been suggested that all variables that are to appear in complete data model, all variables that appear in the response model, and all variables that explain a large degree of the variance be included in the imputation model. Additionally, removal of those covariates with too much missingness has also been suggested.[138] Other methods for variable selection, however have not been explored.

In this study, we operationalized the selection of investigator-specified covariates by including all of the covariates, including the exposure and outcomes, that were obtained for the chronic HCV and incident acute MI study which was also conducted in THIN. Of note, if patients had ever had diagnostic codes for medical conditions or ever received a prescription for a medication used to classify a patient as having such a condition (i.e. insulin for treatment of diabetes) at any time during their period of observation, they were considered to have that condition/ attribute for purposes of this analysis.

High Dimensional Variable Selection

A high dimensional approach to propensity score modeling has been proposed by Schneiweiss and colleagues to adjust for multiple confounders in claims data.[139] This approach to selection is based on the theory that a set of proxies, identifiable from information available in claims or clinical databases, reflects the global health state of the patient in question. These proxies may consist of different levels of information including symptoms, diagnoses, physician prescribing practices, dispensing of prescriptions and receipt of medications. All of these facets of care translate into observations made about the patient, observations that affect the questions being

explored in electronic health record data. Additionally, and perhaps more importantly, such proxies may be important in adjusting for confounders that are unobserved.

The method as adapted for this study included five of the seven steps identified for determining the importance of covariates for creating a propensity score model, the initial setting in which this approach was proposed and validated. The five steps utilized included the following: 1. Identification of the data source's data dimensions; 2. Identification of candidate covariates based on prevalence of entry into the database; 3. Assessment of occurrence, or frequency, of codes in each data dimension; 4. Prioritization of covariates; and 5. Selection of covariates. To operationalize this approach, we used the internal structure of the THIN Read codes and British National Formulary (BNF) codes and separated them up to their third and fourth digits/characters, respectively. Given that some of the codes were available in only a few subjects, rather than using prevalence, we included variables based on their correlation with BMI (Pearson correlation coefficient > 0.1, *a priori* statistical threshold set).

Machine Learning

Machine learning (ML) is a form of artificial intelligence that focuses on a computer's ability to recognize patterns so that it can "learn.[140] It is intricately linked to but often confused with data mining, which concentrates on the discovery of data within a system that were previously unknown. Computational learning theory is the form of computer science under which machine learning algorithms are used to make predictions about data. Because the future is unknown, ML can make no claims about the certainty of the predictions but can inform the probabilities of performance. Many algorithms exist for ML including decision trees, association rules, linear models, instance-based learning, multi

instance-based learning and cluster analysis. In this protocol, the predictive ability of classification and decision trees (CART) and linear regression were examined in the context of multiple imputation.

Classification and Decision Tree Analysis (CART)

The construction of a decision trees is based on recursive partitioning of data. Nodes are formed by the testing of attributes, or variables, within instances, or observations, and then compared to a constant. Leaf nodes are then created to house classifications of instances within the data that flow down the tree. Regression trees are formed when the numeric outputs form the nodes.

The choice of attribute for any given node is dependent upon the amount of information that can be specified by the use of this attribute at a branch point. This quantity is referred to as information gain. For all scenarios, the quantity "entropy" determines the information gained.

$$\textbf{Entropy} = (p_1, p_2 \ldots p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

Given that attributes with many possible values dominate entropy, the gain ratio (total gain in tree/ entropy of attribute) can be used in place of entropy to prioritize nodes.

Linear regression

In linear regression, weights are determined for important attributes as they are in standard linear regression used in statistics. The same is true of logistic regression though hyperplanes can be constructed instead and classification determined based on where the instances in the data lie in space with respect to these hyperplanes.

Operationalizing use of Machine Learning Algorithms

Data that were structured for the high dimensional variable selection approach were also utilized for the machine learning algorithms. Given the large volume of data, with datasets often including more than 5,000 variables, other techniques, such as random forest, were entertained but ultimately abandoned given the time required for computation (simulations required approximately 1-2 weeks for generation of results pending server availability).

Specification of Multiple Imputation Models

For all of the approaches to variable selection, multiple imputation was then undertaken as proposed by the Rubin method. In brief, multiple complete datasets (n=10) were generated and within each dataset missing values of BMI were replaced with values that reflect patterns of complete and incomplete and independent and dependent covariates. Though Rubin's method also includes the creation of a response model in which point estimates and their adjusted variances are then pooled using standard procedures to generate a summary measure of effect,[122] this was not performed as part of this simulation. Instead, the accuracy of the prediction using the true BMI as the gold standard was determined by calculating the percent bias and mean squared error estimates and percent correctly classified, with BMI categorized as < 30 kg/m^2 and ≥ 30 kg/m^2, and the 3 approaches to variable selection were compared.

Of note, there are several ways to specify the execution of multiple imputation within statistical packages. Regarding the specification of prediction of a continuous variable, a simple regression model, a multivariate normal regression or use of chained equations may be employed. Each of these methods, while related, provided a slightly different

estimate of percent bias and mean squared error (**Table 4.1**). Therefore, because the use of chained equations requires knowledge of the underlying distribution of the missing variables for correct specification and because the regress command does not allow for imputation of multiple missing variables, a multivariate normal approach was used for all simulations.

Table 4.1: Comparison of bias of statistical approaches for specification of multiple imputation models

Multiple Imputation Specification	MCAR	MAR				MNAR			
		1	2	3	4	1	2	3	4
Chained Equations*	3.09% (2.81 - 3.36)	3.03% (2.70 - 3.36)	3.16% (2.86 - 3.47)	3.34% (2.95 - 3.73)	3.29% (3.00 - 3.58)	-1.26% (1.37 - 1.14)	21.66% (-21.78 - -21.54)	19.15% (18.96 - 19.34)	6.47% (6.31 - 6.63)
Chained Equations^	3.07% (2.79 - 3.34)	3.03% (2.76 – 3.29)	3.20% (2.93 - 3.46)	3.26% (2.88 – 3.64)	3.34% (1.32 - 3.64)	-1.25% (-1.39 - -1.11)	-21.65% (-21.77- - 21.54)	19.26% (19.02 - 19.51)	6.88% (6.33 – 6.64)
Multivariate Normal*	3.03% (2.75 - 3.31)	3.04% (2.75 - 3.32)	3.17% (3.14 - 3.20)	3.33% (2.95 - 3.71)	3.34% (3.05 - 3.62)	-1.24% (-1.32 - -1.15)	-21.59% (-21.74 - -21.44)	19.20% (19.00 - 19.41)	6.49% (6.35 - 6.64)
Multivariate Normal^	3.07% (2.82 - 3.32)	3.08% (2.80 -3.36)	3.15% (2.86 - 3.43)	3.29% (2.88 - 3.70)	3.30% (3.03 - 3.57)	-1.23% (-1.36 - -1.12)	-21.62% (-21.73 - -21.52)	19.20% (18.98 - 19.41)	6.48% (6.33 – 6.63)
Regression^	3.07% (2.79 - 3.34)	3.03% (2.76 - 3.29)	3.20% (2.93 -3.46)	3.26% (2.88 - 3.64)	3.34% (3.05 - 3.64)	-1.25% (-1.39 - -1.11)	-21.65% (-21.77 - -21.54)	19.26% (19.02 - 19.51)	6.49% (6.33 – 6.64)

*BMI, measure of urban/ rural dwelling and a marker of socioeconomic status (Townsend index) imputed
^BMI only imputed

Assessment of Prediction Methods

After performing 1000 simulations of missingness and specification of the multiple imputation models, the resulting BMI values were compared using estimates of percent bias and standardized mean squared error (MSE). Percent bias was calculated with the following formula: [((BMI predicted) – (BMI truth))/(BMI truth)]. Bias was thereafter summed over the estimates for each of the 1000 simulations to generate a summary estimate. Standardized MSE was calculated using the following formula: [(error)^2/BMI truth]. This procedure was untaken so that percent bias and MSE were reported on the same scale. Standardized MSEs were also summed over all simulations and a summary estimate reported.

We additionally tried to specify the imputation method for prediction of a dichotomous outcome, BMI categorized as < 30 kg/m^2 and ≥ 30 kg/m^2, to assess percent correct classification. However, given the size of the dataset and the finding that there was perfect prediction of classification for many of the Read and BNF codes included (complete separation of the covariates), the performance of the methods using differing approaches to variable selection could not be assessed in this fashion. Instead, once missing values for BMI were generated in a continuous fashion, they were converted to a categorical variable and the percent correctly classified then determined by comparing the generated BMI category to that of BMI truth.

Statistical analyses were conducted in Stata, version 14.2 (StataCorp, College Station, TX) and R, version 3.3.3 (Stats and rpart packages, R Foundation for Statistical Computing, Vienna, Austria).[141]

Results:

The cohort assembled in the parent dataset included 203,622 patients who were enrolled in THIN from 2005 through 2008 and had a determination of BMI during their enrollment. The median age of the cohort was 55 years (interquartile range, IQR 43, 69). Forty-five percent (93,203) were male. The median height was 1.68 meters (IQR 1.61, 1.75) and weight was 75 kilograms (IQR 64, 88). For BMI, the median was 26.2 kg/m^2 (IQR 23.1, 30.0). When characterized in a dichotomous fashion, BMI was greater than or equal to 30 kg/m^2 in 25.5% of the cohort.

Prediction of BMI in a Continuous Fashion

The smallest estimates of bias were observed, for the most part, with the machine learning approaches employed, either CART or linear regression, See tables

4.2, 4.3 and 4.4. In each of the examples of missingness mechanisms explored, the traditional investigator-specified approach produced the most biased estimates, including in the case of the MCAR mechanism. Percent bias for prediction in the simulated MCAR mechanism of missingness ranged from 2.65% (ML approach, linear regression) to 3.75% (traditional investigator-specified approach). The performance patterns were similar in the MAR simulations, with linear regression in machine learning performing most favorably (percent bias ranging from 2.54%, 95% CI 2.37 - 2.71 to 2.98%, 95% CI 2.80 - 3.16) and the traditional investigator-specified approach performing less favorably (percent bias ranging from 3.71%, 95% CI 3.69 - 3.73 to 4.15%, 95%CI 4.13 - 4.17). In the case of the MNAR algorithm, the methods for variable selection performed in the same fashion as that seen with MCAR and MAR, with the ML approaches being less biased. In general, the ML approaches also tended to be more accurate, as evidenced by the smallest standardized MSEs, than the other variable selection methods evaluated.

Of note, all forms of variable selection performed poorly when BMI was simulated as missing based on extreme values (patterns 2 and 3 of the MNAR mechanism of missingness). For essentially all approaches, the estimates of percent bias were in the 16-26% range. This is not unexpected and suggests that when there are no associations between missingness and other observable covariates, prediction with multiple imputation will be biased.

It is noteworthy that all approaches were able to predict MNAR patterns 1 and 4 with similar percent bias and accuracy as assessed by the standardized MSE. Since the pattern simulated was linear in nature in pattern 1 and U shaped in pattern 4, it is

plausible that other variables in the data set were able to improve the prediction of this pattern of missingness, both under specified and unspecified conditions.

Table 4.2: Percent bias and mean squared error results for prediction of BMI for the three approaches to variable selection: Stimulation for data missing completely at random (MCAR)

Variable Selection	Investigator Specified	High Dimensional	Machine Learning	
			CART	Linear Regression
Percent Bias	3.75% (3.73 - 3.77)	3.11% (3.09 - 3.13)	3.33% (3.16 - 3.50)	2.65% (2.48 - 2.83)
Standardized Mean Squared Error	2.20 (2.19 - 2.20)	1.79 (1.79 - 1.79)	0.96 (0.96 - 0.96)	1.22 (1.22 - 1.22)

Table 4.3: Percent bias and mean squared error results for prediction of BMI for the three approaches to variable selection: Stimulation for data missing at random (MAR)

Variable Selection	Investigator Specified	High Dimensional	Machine Learning	
			CART	Linear Regression
Mechanism 1				
Percent Bias	3.71% (3.69 - 3.73)	3.04% (3.03- 3.06)	2.84% (2.67 - 3.02)	2.54% (2.37 - 2.71)
Standardized Mean Squared Error	2.16 (2.15 - 2.16)	1.77 (1.77 - 1.77)	0.91 (0.91 - 0.91)	1.22 (1.22 - 1.22)
Mechanism 2				
Percent Bias	3.80% (3.79 - 3.82)	3.21% (3.19 - 3.23)	3.09% (2.92 - 3.27)	2.61% (2.43 - 2.78)
Standardized Mean Squared Error	2.20 (2.20 - 2.20)	1,78 (1.78 - 1.78)	0.95 (0.95 - 0.96)	1.17 (1.16 - 1.17)
Mechanism 3				
Percent Bias	3.99% (3.97 - 4.01)	3.25% (3.23 - 3.27)	3.61% (3.44 - 3.78)	2.76% (2.59 - 2.92)
Standardized Mean Squared Error	2.20 (2.20 - 2.21)	1.80 (1.80 - 1.80)	0.96 (0.96 - 0.96)	1.28 (1.28 - 1.28)
Mechanism 4				
Percent Bias	4.15% (4.13 - 4.17)	3.37% (3.36 - 3.39)	3.97% (3.79 - 4.15)	2.98% (2.80 - 3.16)
Standardized Mean Squared Error	2.26 (2.25 - 2.26)	1.82 (1.82 - 1.82)	1.03 (1.02 - 1.03)	1.25 (1.25 - 1.25)

Table 4.4: Percent bias and mean squared error results for prediction of BMI for the three approaches to variable selection: Stimulation for data missing not at random (MNAR)

Variable Selection	Investigator Specified	High Dimensional	Machine Learning	
			CART	Linear Regression
Mechanism 1				
Percent Bias	-1.68% (-1.70 - -1.67)	-1.33% (-1.35 - -1.32)	-1.72% (-1.89 - -1.55)	-1.45% (-1.62 - -1.27)
Standardized Mean Squared Error	2.15 (2.15 - 2.15)	1.74 (1.74 -1.74)	1.14 (1.14 - 1.14)	1.33 (1.33 - 1.33)
Mechanism 2				
Percent Bias	-25.55% (-25.56 - -25.54)	-21.58% (-21.59 - -21.57)	-23.08% (-23.27 - -22.88)	-19.69% (-19.88 - -19.49)
Standardized Mean Squared Error	3.28 (3.27 - 3.28)	2.56 (2.56 - 2.56)	2.66 (2.66 - 2.67)	2.34 (2.34 - 2.35)
Mechanism 3				
Percent Bias	22.0% (22.0 - 22.1)	19.19% (19.18 - 19.21)	20.00% (19.81 - 20.19)	16.27% (16.08 - 16.46)
Standardized Mean Squared Error	3.00 (2.99 - 3.00)	2.42 (2.42 - 2.42)	1.20 (1.19 - 1.20)	1.36 (1.36 - 1.36)
Mechanism 4				
Percent Bias	6.79% (6.77 - 6.81)	6.44% (6.42 - 6.46)	6.08% (5.90 - 6.26)	4.90% (4.72 - 5.08)
Standardized Mean Squared Error	2.59 (2.59 - 2.60)	2.12 (2.12 - 2.12)	1.43 (1.43 - 1.43)	1.51 (1.51 - 1.51)

Prediction of BMI in a Dichotomous Fashion

As observed with the prediction of BMI in a continuous fashion, the ML approaches produced the least biased estimates (Table 4.5). Interestingly, percent correctly classified was, except in one instance, highest with the CART approach (79%, 95% CI 79.0 - 79.0 for the MCAR, 78.1%, 95% CI 78.1 - 78.1 to 79.6%, 95% CI 79.6 - 79.6 for the MAR mechanisms and 21.5%, 95% CI 21.5 - 21.5 to 97.4%, 95% CI 97.4 - 97.4 for the MNAR mechanisms). As noted above, the prediction of pattern 2 of the MNAR mechanism of missingness was poor with percent correctly classified ranging

from 14.2% (95% CI 14.1 - 14.2) for the traditional investigator-specified approach to 26.6% (95% CI 26.6 - 26.6) for the high dimensional approach. Unexpectedly, prediction of missing values as simulated under pattern 3 of MNAR (U shaped missingness) was comparable to the percent correctly classified for the MCAR and MNAR mechanisms (Table 4.5).

Table 4.5: Percent correctly classified for dichotomous characterization of BMI

Variable Selection	Investigator Specified	High Dimensional	Machine Learning CART	Machine Learning Linear Regression
MCAR				
Percent Correctly Classified	61.6% (61.6 - 61.7)	67.2% (67.2 - 67.3)	79.0% (79.0 - 79.0)	75.3% (75.3 - 75.3)
MAR				
Simulation Pattern				
1	60.8% (60.8 - 60.8)	66.2% (66.2 - 66.2)	78.8% (78.0 - 78.0)	74.0% (74.0 - 74.0)
2	62.4% (62.4 - 62.4)	67.8% (67.8 - 67.8)	79.5% (79.5 - 79.5)	76.4% (76.4 - 76.4)
3	60.7% (60.7 - 60.8)	66.6% (66.6 - 66.6)	78.1% (78.1 - 78.1)	74.0% (74.0 - 74.0)
4	62.1% (62.1 - 62.1)	68.1% (68.1 - 68.2)	79.6% (79.6 - 79.6)	76.5% (76.5 - 76.5)
MNAR				
Simulation Pattern				
1	59.3% (59.2 - 59.3)	65.3% (65.2 - 65.3)	73.0% (73.0 - 73.0)	71.8% (71.8 - 71.8)
2	14.2% (14.1 - 14.2)	26.0% (25.9 - 26.0)	21.5% (21.5 - 21.5)	25.5% (25.5 - 25.5)
3	65.6% (65.5 - 65.6)	71.6% (71.6 - 71.6)	97.4% (97.4 - 97.4)	86.3% (86.3 - 86.3)
4	62.5% (62.5 - 62.5)	68.6% (68.6 - 68.7)	79.7% (79.7 - 79.7)	77.0% (77.0 - 77.0)

Discussion:

In this simulation study in which different approaches to variable selection and their performance in the prediction of BMI were evaluated in multiple imputation, we found that, for the most part, machine learning algorithms provided the least biased estimates and hence best prediction of BMI. However, we caution that the relative differences in percent bias were generally small, sometimes much less than 1%, and in the context of correct specification of the response model in multiple imputation may increase bias only marginally. Additionally, these techniques were found to be computationally onerous, at times requiring weeks to run high dimensional and ML analyses. While investigators employing such strategies for their respective studies would not have to engage in the simulation portion of this statistical exercise, additional processing time would undoubtedly be required for any analysis.

For simplicity of the discussion, MCAR, the first pattern of missingness for MAR and the first pattern of missingness for MCAR will be discussed primarily though all results are presented for review. Firstly, though for MCAR the percent bias was fairly low, in the 2.65%-3.75% range, these data did represent a simple random sample of the parent dataset and hence the percent bias would be expected to be closer to zero. The second observation of interest is that estimates of bias seemed to decrease as more complex structures of missingness were applied. While contrary to what was hypothesized prior to implementing the simulation (good to great prediction of MCAR and MAR data), a new hypothesis is that the machine learning approaches were able to interpret the inherent patterns within the simulated mechanisms of missingness. This may be on the basis of the algorithm's ability to incorporate an exhaustive number of potential variables into its prediction as well as the lack of specification required for the generation of its estimates. Pattern 1 of MNAR is a pattern of missingness that is constructed upon a linear relationship between increasing BMI and missingness. It is

possible that the ML algorithms were able to detect this pattern of missingness from other variables and learned the frame for the missingness simulation. Lastly, these methods were qualitatively more accurate for the prediction of BMI in a continuous fashion rather than when BMI was specified as a categorical variable. However, there is no direct way to compare percent correctly categorized as a categorical variable with percent bias when measuring a continuous variable. Moreover, the qualitative assessment may in part be due to measurement of percent bias without considering the absolute difference between estimated BMI and truth.

The programming time and computer processing time required for completion of the 1000 simulations was substantial, sometimes requiring weeks for completion of a single analysis. While we endorse that the prediction afforded by using some of the more novel strategies such as the machine learning approaches was numerically better for prediction, the robustness of the results need to be weighed against the time required for the statistical tasks related to the project of interest.

Little guidance is available in the literature on correct model specification for multiple imputation models, as generated for the prediction of missing data. Correct model specification and methods for appropriate variable selection in multiple imputation are required to ensure that the most accurate measure of association can be determined. As BMI is an important potential confounder for the study of metabolic diseases, cardiovascular disease and cancer, and missing BMI data was an issue encountered in the second aim of this dissertation, prediction of BMI was undertaken for this simulation study. However, the results of this methodologic exploration are applicable to any variable, including the exposure or the outcome variable in an epidemiologic study.

These data will contribute to the small body of literature on how investigators choose variables for inclusion in multiple imputation models. It is also our hope that the characterization of THIN's hierarchy and extraction of data in its organized form will be of assistance to researchers in the future who may be unsure of the variables of interest for prediction of variables in THIN and may be able to use this automated approach.

CONCLUSIONS

Chronic HCV infection is a complex disease process characterized by both hepatic and extrahepatic manifestations that increase its morbidity and mortality. Additionally, chronic HCV affects the quality of life of those affected and decreases their productivity. Given the high prevalence of chronic HCV infection and its personal, financial and societal burden, chronic HCV will remain a public health burden until its eradication. Fortunately, with the advent and approval of DAA therapy, eradication of HCV infection may be realized in the next several decades.

This dissertation explored 2 of chronic HCV infection's extrahepatic manifestations; the first, its effect on cholesterol metabolism and the second, its effect on cardiovascular disease. In the first aim, we determined that while patients with chronic HCV infection have relatively low total cholesterol, LDL and triglycerides, treatment with statins disrupts not only cholesterol metabolism but was also associated with lower HCV RNA, an apparent treatment effect that was exploited in the prior era of interferon based therapy to try to boost response rates. Statins also have other pleiotropic effects and the finding of a low rate of aminotransferase elevation in the setting of statin therapy suggests that there are few downsides to using this therapy. This is particularly relevant in this patient population who is also at increased risk for diabetes mellitus, a highly significant risk factor for cardiovascular disease.

In the second aim, we explored the association between chronic HCV infection and cardiovascular disease. While we found no association between chronic HCV infection and incident acute MI, we suspect that the limitations of our study such as not

including other forms of cardiac disease, relatively short follow up time and a small sample of HCV infected patients could have masked a true association. Other groups have published on the topic and have observed a modest increase in cardiovascular morbidity and mortality in chronic HCV infection. As the mechanism of the development of atherosclerotic disease and HCV share a common root in the role of the immune system, it remains an important task to understand the associations between chronic systemic infections and the biology of cardiovascular disease. This should continue to be explored both on an epidemiologic and a translational basis during the coming years as we hopefully are able to successfully eradicate the virus.

Lastly, we performed a methodologic study examining variable selection approaches for specification of multiple imputation models. We found that while machine learning techniques provided the least biased estimates, these methods were computationally onerous without offering marked improvement in bias estimates. It is our sincerest hope that these three studies have addressed knowledge gaps in the field and their methodology serve as a resource for other epidemiologic researchers.

APPENDIX

Appendix Table 3.1. Read diagnostic codes used to identify patients with hepatitis C virus (HCV) infection in the THIN database.

Read Diagnostic Code	Disease Classification
A704000	Viral hepatitis C with coma
A705000	Viral hepatitis C without hepatic coma
A707200	Chronic viral hepatitis C
A70z000	Hepatitis C
ZV02C00	Hepatitis C carrier
2J11.00	Hepatitis C immune
43X3.00	Hepatitis C antibody test positive
65Q7.00	Viral hepatitis carrier with GP comment of 'hepatitis C'
A705400	Hepatitis non-A, non-B
2J1..00	Hepatitis C status

Appendix Table 3.2. Read diagnostic codes used to identify patients with viral hepatitis not otherwise specified in the THIN database.

Read Diagnostic Code	Disease Classification
A70.00	Viral hepatitis
A704.00	Other specified viral hepatitis with coma
A704z00	Other specified viral hepatitis with hepatic coma NOS
A705.00	Other specified viral hepatitis without coma
A705z00	Other specified viral hepatitis without mention of coma NOS
A706.00	Unspecified viral hepatitis with coma
A707.00	Chronic viral hepatitis
A707X00	Chronic viral hepatitis, unspecified
A70z.00	Unspecified viral hepatitis
AyuB.00	Viral hepatitis
AyuB000	Other specified acute viral hepatitis
AyuB100	Other chronic viral hepatitis
AyuB200	Chronic viral hepatitis, unspecified
AyuB300	Unspecified viral hepatitis with coma
AyuB400	Unspecified viral hepatitis without coma
J614.00	Chronic hepatitis
J614000	Chronic persistent hepatitis
J614100	Chronic active hepatitis
J614200	Chronic aggressive hepatitis
J614y00	Chronic hepatitis unspecified
J614z00	Chronic hepatitis NOS
J631.00	Hepatitis in viral diseases EC
J631z00	Hepatitis in viral diseases EC NOS
J633.00	Hepatitis unspecified

J633z00	Hepatitis unspecified NOS

Appendix Table 3.3. Read diagnostic codes used to identify patients with an acute myocardial infarction (MI) in the THIN database.

Read Diagnostic Code	Disease Classification
323..00	ECG: myocardial infarction
G30X.00	Acute transmural myocardial infarction of unspecified site
G361.00	Atrial septal defect/current comp follow acute myocardial infarction
G361.00	Atrial septal defect/current comp follow acute myocardial infarction
G362.00	Ventric septal defect/current comp follow acute myocardial infarction
G362.00	Ventric septal defect/current comp follow acute myocardial infarction
14A4.00	H/O: myocardial infarct >60
3234	ECG: posterior/inferior infarction
G304.00	Posterior myocardial infarction NOS
G308.00	Inferior myocardial infarction NOS
G30y200	Acute septal infarction
G366.00	Thrombosis atrium
G366.00	Thrombosis atrium
G307.00	Acute subendocardial infarction
G34y100	Chronic myocardial ischaemia
G360.00	Haemopericardium/current comp follow acute myocardial infarction
G360.00	Haemopericardium/current comp follow acute myocardial infarction
G305.00	Lateral myocardial infarction NOS
G30..15	MI - acute myocardial infarction
G300.00	Acute anterolateral infarction
G344.00	Silent myocardial ischaemia
G38..00	Postoperative myocardial infarction
G302.00	Acute inferolateral infarction
G303.00	Acute inferoposterior infarction

3235	ECG: subendocardial infarction
G301.00	Other specified anterior myocardial infarction
G301000	Acute anteroapical infarction
G31y200	Subendocardial ischaemia
G5y1.00	Myocardial degeneration
322..00	ECG: myocardial ischaemia
322Z.00	ECG: myocardial ischaemia NOS
G30..17	Silent myocardial infarction
14A3.00	H/O: myocardial infarct <60
G381.00	Postoperative transmural myocardial infarction inferior wall
G306.00	True posterior myocardial infarction
G30..00	Acute myocardial infarction
G30z.00	Acute myocardial infarction NOS
G32..12	Personal history of myocardial infarction
G350.00	Subsequent myocardial infarction of anterior wall

Appendix Table 3.4. Read diagnostic codes used to identify patients with a coronary intervention in the THIN database.

Read Diagnostic Code	Disease Classification
792..11	Coronary artery bypass graft operations
SP07600	Coronary artery bypass graft occlusion
ZV45K00	[V]Presence of coronary artery bypass graft
ZV45K11	[V]Presence of coronary artery bypass graft – CABG
792..00	Coronary artery operations
792..11	Coronary artery bypass graft operations
7920.00	Saphenous vein graft replacement of coronary artery
7920.11	Saphenous vein graft bypass of coronary artery
7920000	Saphenous vein graft replacement of one coronary artery
7920100	Saphenous vein graft replacement of two coronary arteries
7920200	Saphenous vein graft replacement of three coronary arteries
7920300	Saphenous vein graft replacement of four+ coronary arteries
7920y00	Saphenous vein graft replacement of coronary artery OS
7920z00	Saphenous vein graft replacement coronary artery NOS
7921.00	Other autograft replacement of coronary artery
7921.11	Other autograft bypass of coronary artery
7921000	Autograft replacement of one coronary artery NEC
7921100	Autograft replacement of two coronary arteries NEC
7921200	Autograft replacement of three coronary arteries NEC
7921300	Autograft replacement of four of more coronary arteries NEC
7921y00	Other autograft replacement of coronary artery OS
7921z00	Other autograft replacement of coronary artery NOS
7922.00	Allograft replacement of coronary artery
7922.11	Allograft bypass of coronary artery

7922000	Allograft replacement of one coronary artery
7922100	Allograft replacement of two coronary arteries
7922200	Allograft replacement of three coronary arteries
7922300	Allograft replacement of four or more coronary arteries
7922y00	Other specified allograft replacement of coronary artery
7922z00	Allograft replacement of coronary artery NOS
7923.00	Prosthetic replacement of coronary artery
7923.11	Prosthetic bypass of coronary artery
7923000	Prosthetic replacement of one coronary artery
7923100	Prosthetic replacement of two coronary arteries
7923200	Prosthetic replacement of three coronary arteries
7923300	Prosthetic replacement of four or more coronary arteries
7923y00	Other specified prosthetic replacement of coronary artery
7923z00	Prosthetic replacement of coronary artery NOS
7924.00	Revision of bypass for coronary artery
7924000	Revision of bypass for one coronary artery
7924100	Revision of bypass for two coronary arteries
7924200	Revision of bypass for three coronary arteries
7924300	Revision of bypass for four or more coronary arteries
7924400	Revision of connection of thoracic artery to coronary artery
7924y00	Other specified revision of bypass for coronary artery
7924z00	Revision of bypass for coronary artery NOS
7925.00	Connection of mammary artery to coronary artery
7925.11	Creation of bypass from mammary artery to coronary artery
7925000	Double anastomosis of mammary arteries to coronary arteries
7925100	Double implant of mammary arteries into coronary arteries
7925200	Single anast mammary art to left ant descend coronary art
7925300	Single anastomosis of mammary artery to coronary artery NEC

7925400	Single implantation of mammary artery into coronary artery
7925y00	Connection of mammary artery to coronary artery OS
7925z00	Connection of mammary artery to coronary artery NOS
7926.00	Connection of other thoracic artery to coronary artery
7926000	Double anastom thoracic arteries to coronary arteries NEC
7926100	Double implant thoracic arteries into coronary arteries NEC
7926200	Single anastomosis of thoracic artery to coronary artery NEC
7926300	Single implantation thoracic artery into coronary artery NEC
7926y00	Connection of other thoracic artery to coronary artery OS
7926z00	Connection of other thoracic artery to coronary artery NOS
7927.00	Other open operations on coronary artery
7927500	Open angioplasty of coronary artery
7927y00	Other specified other open operation on coronary artery
7927z00	Other open operation on coronary artery NOS
7928.00	Transluminal balloon angioplasty of coronary artery
7928.11	Percutaneous balloon coronary angioplasty
7928000	Percut transluminal balloon angioplasty one coronary artery
7928100	Percut translum balloon angioplasty mult coronary arteries
7928200	Percut translum balloon angioplasty bypass graft coronary a
7928300	Percut translum cutting balloon angioplasty coronary artery
7928y00	Transluminal balloon angioplasty of coronary artery OS
7928z00	Transluminal balloon angioplasty of coronary artery NOS
7929.00	Other therapeutic transluminal operations on coronary artery
7929000	Percutaneous transluminal laser coronary angioplasty
7929100	Percut transluminal coronary thrombolysis with streptokinase
7929111	Percut translum coronary thrombolytic therapy- streptokinase
7929200	Percut translum inject therap subst to coronary artery NEC
7929300	Rotary blade coronary angioplasty

7929400	Insertion of coronary artery stent
7929500	Insertion of drug-eluting coronary artery stent
7929600	Percutaneous transluminal atherectomy of coronary artery
7929y00	Other therapeutic transluminal op on coronary artery OS
7929z00	Other therapeutic transluminal op on coronary artery NOS
792A.00	Diagnostic transluminal operations on coronary artery
792B.00	Repair of coronary artery NEC
792B000	Endarterectomy of coronary artery NEC
792B100	Repair of rupture of coronary artery
792By00	Other specified repair of coronary artery
792Bz00	Repair of coronary artery NOS
792C.00	Other replacement of coronary artery
792C000	Replacement of coronary arteries using multiple methods
792Cy00	Other specified replacement of coronary artery
792Cz00	Replacement of coronary artery NOS
792D.00	Other bypass of coronary artery
792Dy00	Other specified other bypass of coronary artery
792Dz00	Other bypass of coronary artery NOS
792y.00	Other specified operations on coronary artery
792z.00	Coronary artery operations NOS
793G.00	Perc translumin balloon angioplasty stenting coronary artery
793Gy00	OS perc translumina balloon angioplast stenting coronary art
793Gz00	Perc translum balloon angioplasty stenting coronary art NOS
ZV45700	[V]Presence of aortocoronary bypass graft
ZV45800	[V]Presence of coronary angioplasty implant and graft
ZV45K00	[V]Presence of coronary artery bypass graft
ZV45K11	[V]Presence of coronary artery bypass graft – CABG
ZV45L00	[V]Status following coronary angioplasty NOS

792..11	Coronary artery bypass graft operations
SP07600	Coronary artery bypass graft occlusion
ZV45K00	[V]Presence of coronary artery bypass graft
ZV45K11	[V]Presence of coronary artery bypass graft – CABG
792..00	Coronary artery operations
792..11	Coronary artery bypass graft operations
7920.00	Saphenous vein graft replacement of coronary artery
7920.11	Saphenous vein graft bypass of coronary artery
7920000	Saphenous vein graft replacement of one coronary artery
7920100	Saphenous vein graft replacement of two coronary arteries
7920200	Saphenous vein graft replacement of three coronary arteries
7920300	Saphenous vein graft replacement of four+ coronary arteries
7920y00	Saphenous vein graft replacement of coronary artery OS
7920z00	Saphenous vein graft replacement coronary artery NOS
7921.00	Other autograft replacement of coronary artery
7921.11	Other autograft bypass of coronary artery
7921000	Autograft replacement of one coronary artery NEC
7921100	Autograft replacement of two coronary arteries NEC
7921200	Autograft replacement of three coronary arteries NEC
7921300	Autograft replacement of four of more coronary arteries NEC
7921y00	Other autograft replacement of coronary artery OS
7921z00	Other autograft replacement of coronary artery NOS
7922.00	Allograft replacement of coronary artery
7922.11	Allograft bypass of coronary artery
7922000	Allograft replacement of one coronary artery
7922100	Allograft replacement of two coronary arteries
7922200	Allograft replacement of three coronary arteries
7922300	Allograft replacement of four or more coronary arteries

7922y00	Other specified allograft replacement of coronary artery
7922z00	Allograft replacement of coronary artery NOS
7923.00	Prosthetic replacement of coronary artery
7923.11	Prosthetic bypass of coronary artery
7923000	Prosthetic replacement of one coronary artery
7923100	Prosthetic replacement of two coronary arteries
7923200	Prosthetic replacement of three coronary arteries
7923300	Prosthetic replacement of four or more coronary arteries
7923y00	Other specified prosthetic replacement of coronary artery
7923z00	Prosthetic replacement of coronary artery NOS
7924.00	Revision of bypass for coronary artery
7924000	Revision of bypass for one coronary artery
7924100	Revision of bypass for two coronary arteries
7924200	Revision of bypass for three coronary arteries
7924300	Revision of bypass for four or more coronary arteries
7924400	Revision of connection of thoracic artery to coronary artery
7924y00	Other specified revision of bypass for coronary artery
7924z00	Revision of bypass for coronary artery NOS
7925.00	Connection of mammary artery to coronary artery
7925.11	Creation of bypass from mammary artery to coronary artery
7925000	Double anastomosis of mammary arteries to coronary arteries
7925100	Double implant of mammary arteries into coronary arteries
7925200	Single anast mammary art to left ant descend coronary art
7925300	Single anastomosis of mammary artery to coronary artery NEC
7925400	Single implantation of mammary artery into coronary artery
7925y00	Connection of mammary artery to coronary artery OS
7925z00	Connection of mammary artery to coronary artery NOS
7926.00	Connection of other thoracic artery to coronary artery

7926000	Double anastom thoracic arteries to coronary arteries NEC
7926100	Double implant thoracic arteries into coronary arteries NEC
7926200	Single anastomosis of thoracic artery to coronary artery NEC
7926300	Single implantation thoracic artery into coronary artery NEC
7926y00	Connection of other thoracic artery to coronary artery OS
7926z00	Connection of other thoracic artery to coronary artery NOS
7927.00	Other open operations on coronary artery
7927500	Open angioplasty of coronary artery
7927y00	Other specified other open operation on coronary artery
7927z00	Other open operation on coronary artery NOS
7928.00	Transluminal balloon angioplasty of coronary artery
7928.11	Percutaneous balloon coronary angioplasty
7928000	Percut transluminal balloon angioplasty one coronary artery
7928100	Percut translum balloon angioplasty mult coronary arteries
7928200	Percut translum balloon angioplasty bypass graft coronary a
7928300	Percut translum cutting balloon angioplasty coronary artery
7928y00	Transluminal balloon angioplasty of coronary artery OS
7928z00	Transluminal balloon angioplasty of coronary artery NOS
7929.00	Other therapeutic transluminal operations on coronary artery
7929000	Percutaneous transluminal laser coronary angioplasty
7929100	Percut transluminal coronary thrombolysis with streptokinase
7929111	Percut translum coronary thrombolytic therapy- streptokinase
7929200	Percut translum inject therap subst to coronary artery NEC
7929300	Rotary blade coronary angioplasty
7929400	Insertion of coronary artery stent
7929500	Insertion of drug-eluting coronary artery stent
7929600	Percutaneous transluminal atherectomy of coronary artery
7929y00	Other therapeutic transluminal op on coronary artery OS

7929z00	Other therapeutic transluminal op on coronary artery NOS
792A.00	Diagnostic transluminal operations on coronary artery
792B.00	Repair of coronary artery NEC
792B000	Endarterectomy of coronary artery NEC
792B100	Repair of rupture of coronary artery
792By00	Other specified repair of coronary artery
792Bz00	Repair of coronary artery NOS
792C.00	Other replacement of coronary artery
792C000	Replacement of coronary arteries using multiple methods
792Cy00	Other specified replacement of coronary artery
792Cz00	Replacement of coronary artery NOS
792D.00	Other bypass of coronary artery
792Dy00	Other specified other bypass of coronary artery
792Dz00	Other bypass of coronary artery NOS
792y.00	Other specified operations on coronary artery
792z.00	Coronary artery operations NOS
793G.00	Perc translumin balloon angioplasty stenting coronary artery
793Gy00	OS perc translumina balloon angioplast stenting coronary art
793Gz00	Perc translum balloon angioplasty stenting coronary art NOS
ZV45700	[V]Presence of aortocoronary bypass graft
ZV45800	[V]Presence of coronary angioplasty implant and graft
ZV45K00	[V]Presence of coronary artery bypass graft
ZV45K11	[V]Presence of coronary artery bypass graft – CABG
ZV45L00	[V]Status following coronary angioplasty NOS
792..11	Coronary artery bypass graft operations
SP07600	Coronary artery bypass graft occlusion
ZV45K00	[V]Presence of coronary artery bypass graft
ZV45K11	[V]Presence of coronary artery bypass graft – CABG

Appendix Item 4.1: STATA Do-File, Preparation of Dataset/Calculation of Percent Bias

```
/****PhD PROJECT 3: ASSESSMENT OF FEATURE SELECTION STRATEGIES FOR MULTIPLE
IMPUTATION/ VARIABLE SELECTION STRATEGY 1***

***GENERATION OF COMORBIDITY DATA FROM THE MEDICAL FILE IN THIN BASED ON
PROJECT 2 COVARIATES
clear
use "/Users/kforde/Desktop/THIN/code_bmi_dataset.dta"
keep pracid patid bmi_date
sort pracid patid bmi_date
save "/Users/kforde/Desktop/THIN/comorbid_set_up.dta", replace

clear
use "/Users/kforde/Desktop/THIN/Medical_p10.dta"
merge m:1 pracid patid using "/Users/kforde/Desktop/THIN/comorbid_set_up.dta"
keep if _merge==3
drop _merge
drop enddate datatype medflag staffid source episode nhsspec locate textid
category priority medinfo inprac private medid cnsultid modified dteflag
sysdate
***drop if bmi_date< evntdate & evntdate!=.
drop if evntdate==.
***5,776 observations dropped
drop evntdate
***COMORBIDITIES (MED TABLE)************************************************
gen comorbid=0
***HYPERTENSION AS DEFINED BY READ CODES************************************
#delimit ;
foreach x in 14A2.00 246M.00 8BL0.00 2126100 662..12 6627.00 6628.00 6629.00
662B.00 662C.00 662F.00 662G.00 662H.00 662O.00 662P.00 8B26.00 8HT5.00 9N03.00
9N1y200 9OI..00 9OI..11 9OI1.00 9OI2.00 9OI4.00 9OI5.00 9OI6.00 9OI7.00 9OI8.00
9OI9.00 9OIA.00 9OIA.11 9OIZ.00 F404200 F421300 G2...00 G2...11 G20..00 G20..11
G200.00 G201.00 G202.00 G20z.00 G20z.11 G21..00 G210.00 G210000 G210100 G210z00
G211.00 G211000 G211100 G211z00 G21z.00 G21z000 G21z011 G21z100 G21zz00 G22..00
G220.00 G221.00 G222.00 G22z.00 G22z.11 G23..00 G230.00 G231.00 G232.00 G233.00
G23z.00 G24..00 G240.00 G240000 G240z00 G241.00 G241000 G241z00 G241z00 G244.00
G24z.00 G24z000 G24z100 G24zz00 G2y..00 G2z..00 G672.00 G672.11 Gyu2.00 L128.00
SLC6.00 SLC6z00 SyuFT00 TJC7.00 TJC7z00 U60C511 U60C51A {;
qui replace comorbid=1 if medcode=="`x'";
};
***DIABETES AS DEFINED BY READ CODES****************************************;
foreach x in 13AB.00 13AC.00 13B1.00 1434.00 14F4.00 2BBF.00 2G51000 2G5A.00
2G5B.00 2G5C.00 3882.00 3883.00 42c..00 42W..   42W..00 42WZ.00 66A..00 66A1.00
66A2.00 66A3.00 66A4.00 66A5.00 66A8.00 66A9.00 66AA.11 66AD.00 66AG.00 66AH.00
66AI.00 66AJ.00 66AJ.11 66AJ100 66AJz00 66AK.00 66AL.00 66AM.00 66AN.00 66AO.00
66AP.00 66AQ.00 66AR.00 66AS.00 66AT.00 66AZ.00 68A7.00 8A12.00 8A13.00 8CA4100
8H2J.00 8H3O.00 8H4F.00 8H7c.00 8H7F.00 8HKE.00 8HLE.00 8HME.00 8HVU.00 9N1Q.00
9N1V.00 9NM0.00 9OL..00 9OL..11 9OL1.00 9OL2.00 9OL3.00 9OL4.00 9OL5.00 9OL6.00
9OL7.00 9OL8.00 9OLA.00 9OLA.11 9OLZ.00 C10..00 C100.00 C100000 C100011 C100100
C100111 C100112 C100z00 C101.00 C101000 C101100 C101y00 C101z00 C102.00 C102000
C102100 C102z00 C103.00 C103000 C103100 C103y00 C103z00 C104.00 C104.11 C104000
C104y00 C104z00 C105.00 C105000 C105100 C105y00 C105z00 C106.00 C106.11 C106.12
C106.13 C106000 C106100 C106y00 C106z00 C107.00 C107.11 C107.12 C107000 C107100
C107200 C107300 C107400 C107y00 C107z00 C108.00 C108.11 C108.12 C108.13 C108000
C108011 C108012 C108100 C108111 C108112 C108200 C108211 C108212 C108300 C108311
C108312 C108400 C108411 C108412 C108500 C108511 C108512 C108600 C108611 C108612
C108700 C108711 C108712 C108800 C108811 C108812 C108900 C108911 C108912 C108A00
```

```
C108A11 C108A12 C108B00 C108B11 C108B12 C108C00 C108C11 C108C12 C108D00 C108D11
C108D12 C108E00 C108E11 C108E12 C108F00 C108F11 C108F12 C108G00 C108G11 C108G12
C108H00 C108H11 C108H12 C108J00 C108J11 C108J12 C108y00 C108z00 C109.00 C109.11
C109.12 C109.13 C109000 C109011 C109012 C109100 C109111 C109112 C109200 C109211
C109212 C109300 C109311 C109312 C109400 C109411 C109412 C109500 C109511 C109512
C109600 C109611 C109612 C109700 C109711 C109712 C109800 C109900 C109911 C109912
C109A00 C109A11 C109A12 C109B00 C109B11 C109B12 C109C00 C109C11 C109C12 C109D00
C109D11 C109D12 C109E00 C109E11 C109E12 C109F00 C109F11 C109F12 C109G00 C109G11
C109G12 C109H00 C109H11 C109H12 C109J00 C109J11 C109J12 C109K00 C10A.00 C10A000
C10A100 C10A200 C10A300 C10A400 C10A500 C10A600 C10A700 C10AW00 C10AX00 C10B.00
C10B000 C10C.00 C10C.11 C10C.12 C10D.00 C10D.11 C10E.00 C10E.11 C10E.12 C10E000
C10E011 C10E012 C10E100 C10E111 C10E112 C10E200 C10E211 C10E212 C10E300 C10E311
C10E312 C10E400 C10E411 C10E412 C10E500 C10E511 C10E512 C10E600 C10E611 C10E612
C10E700 C10E711 C10E712 C10E800 C10E811 C10E812 C10E900 C10E911 C10E912 C10EA00
C10EA11 C10EA12 C10EB00 C10EB11 C10EB12 C10EC00 C10EC11 C10EC12 C10ED00 C10ED11
C10ED12 C10EE00 C10EE11 C10EE12 C10EF00 C10EF11 C10EF12 C10EG00 C10EG11 C10EG12
C10EH00 C10EH11 C10EH12 C10EJ00 C10EJ11 C10EJ12 C10EK00 C10EK11 C10EL00 C10EL11
C10EM00 C10EM11 C10EN00 C10EN11 C10EP00 C10EP11 C10EQ00 C10F.00 C10F.11 C10F000
C10F011 C10F100 C10F111 C10F200 C10F211 C10F300 C10F311 C10F400 C10F411 C10F500
C10F511 C10F600 C10F611 C10F700 C10F711 C10F800 C10F811 C10F900 C10F911 C10FA00
C10FA11 C10FB00 C10FB11 C10FC00 C10FC11 C10FD00 C10FD11 C10FE00 C10FE11 C10FF00
C10FF11 C10FG00 C10FG11 C10FH00 C10FH11 C10FJ00 C10FJ11 C10FK00 C10FL00 C10FL11
C10FM00 C10FM11 C10FN00 C10FN11 C10FP00 C10FP11 C10FQ00 C10FQ11 C10FR00 C10G.00
C10G000 C10H.00 C10H000 C10J.00 C10J000 C10K.00 C10K000 C10L.00 C10L000 C10M.00
C10M000 C10N.00 C10N000 C10y.00 C10y000 C10y100 C10yy00 C10yz00 C10z.00 C10z000
C10z100 C10zy00 C10zz00 C11y000 C135.00 C135.12 C314.11 C350011 Cyu2.00 Cyu2000
Cyu2100 Cyu2200 Cyu2300 F171100 F345000 F35z000 F372.00 F372.11 F372.12 F372000
F372100 F372200 F381300 F381311 F3y0.00 F420.00 F420000 F420100 F420200 F420300
F420400 F420500 F420z00 F440700 F464000 G73y000 K01x100 K081.00 Kyu0300 L180500
L180600 L180700 L180X00 Lyu2900 M037200 M271000 M271100 M271200 N030000 N030011
N030100 R054200 R054300 TJ23.00 TJ23z00 U602311 U60231E ZC2C800 ZLA2500
ZV65312{;
qui replace comorbid=2 if medcode=="`x'";
};
***HYPERCHOLESTEROLEMIA AS DEFINED BY READ CODES*****************************;
foreach x in 2729H  279 A  279 HL 279 LC 44P2.00
44P3.00 44P4.00 44P5.00 44P6.00 44P7.00 8BAG.00 8BAG000 8BAG100 8BAG200 8CA4700
9N0J.00 C32..11 C320.00 C320.11 C320.12 C320.13 C320000 C320100 C320200 C320300
C320400 C320y00 C320z00 C321.11 C321.12 C322.00 C322.11 C322.13 C44Q2.00 4Q3.00
C320100 C320200 C321.00 C321.11 C321.12 C322.00 C322.11 C322.12 C323.00 C323.11
C323.12 C323.13 C32y400 ZC2CJ00324.00 Cyu8D00 L1420H L1420R  L1420RA U60C600
Y060 CV Y0601JV ZC2CJ00 ZV65317 {;
qui replace comorbid=3 if medcode=="`x'";
};
***RENAL DISEASE AS DEFINED BY READ CODES*************************************;
foreach x in 1Z12.00 1Z13.00 1Z14.00 4412100 44J2.00 44J3000 44J3300 4512.00
4513100 7L1A.11 G222.00 G233.00 G234.00 K05..00 K05..11 K05..12 K050.00 K06..00
K06..11 K060.00 K060.11 K08..00 K08y.00 K08yz00 K08z.00 K0D..00 Kyu2.00 Kyu2100
R144.00 R144.11 SP15400 SP15411 4519.00 7B00.0 7B00100 7B00111 7B00200 7B00211
7B00y00 7B00z00 7B06300 7L1A000 7L1A011 7L1A100 7L1A200 7L1B000 7L1C000 8L50.00
Z91A.00 ZV42000 ZV45100 ZV56.00 ZV56000 ZV56011 ZV56y11 ZV56z00 ZVu3G00 2575.00
7B01511 7L1A.00 7L1Ay00 7L1Az00 7L1B.00 7L1B.11 7L1By00 7L1Bz00 7L1C.00 7L1Cy00
7L1Cz00 C345.00 D215.00 D215000 F374A00 G500400 K080.00 K080z00 K08y100 SP08300
TA02.00 TA02000 TA02011 TA12000 TA12011 TA22000 TA22011 TA42000 TA42011 TB00100
TB00111 TB11.00 TB11.11 U612200 4I29.00 7L1B100 Kyu1C00 SP01500 SP05613 U641.00
Z1A..00 Z1A1.00 Z1A1.11 Z1A2.00 Z1A2.11 Z919.00 Z919100 Z919200 Z919300 Z919400
Z91A100 ZV56100 ZV56y00 2575    4512    K00..12 K02..00 K02..11 K02..12 K020.00
K021.00 K022.00 K023.00 K02y.00 K02y000 K02y100 K02y200 K02y300 K02yz00 K02z.00
K04..00 K040.00 K041.00 K042.00 K04y.00 K04z.00 K08y300 K0A3.00 K0A3000 K0A3100
K0A3200 K0A3300 K0A3400 K0A3500 K0A3600 K0A3700 K0B5.00 K0C0.00 K100.00 K100000
```

```
K100100 K100500 K100z00 Kyu2000 SK05.00 SK05.11 SK08.00 SP15412 SP15413 4519
14S2.0 14V2.00 14V2.11 2728 276 NP  276 RN  403 N   5932EC   5932TB 9956CN
K566 X  K9503AA T924 C310200 C373600 G22..11 J624.00 K010.00 K012.00 K01x000
K01x300 K034.00 K07..00 K07z.00 PD03000 PD04000 SP14300  250 N  2859RF  583 BG
583 EM  583 MN  583 NP  5932A  5932AK  5932AP  5932AT  5932E  5932EA  5932LA
5932MN  5932MP 5932NF 5932RN  7530AP  9977KR  9977KT  14D..11 14D..12 4515.00
8H2M.00 8H3R.00 A160000 C104.00 C104.11 C104000 C104100 C104y00 C104z00 C108000
C109000 C10A200 C314.11 K0...00 K01..00 K011.00 K015.00 K016.00 K017.00 K018.00
K019.00 K01A.00 K01B.00 K01w011 K01wz00 K01x.00 K01x100 K01x200 K01x400 K01xz00
K01y.00 K03..00 K03..11 K03..12 K032600 K03y.00 K03y000 K03yz00 K0A5.00 K0A5000
K0A5100 K0A5200 K0A5600 K0A5700 K0A5X00 K0B..00 K0B1.00 K0B3.00 K0B4000 K0B6.00
K0C1.00 K0C2.00 K0C3.00 K0y..00 K0z..00 C108D00 C109C00 PKy5E00 C108011 C108D11
C109011 C109C11 {;
qui replace comorbid=4 if medcode=="`x'";
};
***OVERWEIGHT AS DEFINED BY READ CODES**************************************;
foreach x in 22K4.00 {;
qui replace comorbid=5 if medcode=="`x'";
};
***OBESITY AS DEFINED BY READ CODES*****************************************;
foreach x in 22K5.00   277 B  277 CP 277 HP  277 HR 1444.00 66C..00 66C1.00
66C2.00 66C4.00 66C5.00 66C6.00 66C7.00 66CE.00 66CZ.00 6878.00 9OK..00 9OK..11
9OK1.00 9OK2.00 9OK3.00 9OK4.00 9OK5.00 9OK6.00 9OK7.00 9OK8.00 9OK9.00 9OKA.00
9OKZ.00 C38..00 C380.00 C380000 C380100 C380200 C380300 C38z.00 C38z000 Cyu7.00
Cyu7000 L161.12 Y060 JY Y0601JY Y0601K4 ZC2CM00 ZV65319 ZV77800 {;
qui replace comorbid=6 if medcode=="`x'";
};
***ALCOHOL*********************************************************************;
foreach x in 136..00 1361.00 1361.11 1361.12 1362.00 1362.11 1362.12 1363.00
1364.00 1365.00 1366.00 1367.00 1368.00 1369.00 136A.00 136B.00 136C.00 136D.00
136E.00 136F.00 136G.00 136H.00 136I.00 136J.00 136K.00 136L.00 136N.00 136O.00
136P.00 136Q.00 136Z.00 1462.00 2577.00 2577.11 8BA8.00 8G32.00 8H35.00 C150500
E01..00 E010.00 E011.00 E011000 E011100 E011z00 E012.00 E012.11 E012000 E013.00
E014.00 E015.00 E01y.00 E01y000 E01yz00 E01z.00 E040.11 E23..00 E23..11 E23..12
E230.00 E230.11 E230000 E230100 E230200 E230300 E230z00 E231.00 E231000 E231100
E231200 E231300 E231z00 E23z.00 E250.00 E250.12 E250.14 E250000 E250100 E250200
E250300 E250z00 Eu10.00 Eu10000 Eu10011 Eu10100 Eu10200 Eu10211 Eu10212 Eu10300
Eu10411 Eu10500 Eu10511 Eu10512 Eu10513 Eu10514 Eu10600 Eu10611 Eu10711 Eu10712
Eu10y00 Eu10z00 F11x000 F144000 F375.00 F394100 G555.00 G852300 J153.00 J610.00
J611.00 J612.00 J612000 J613.00 J613000 J617.00 J617000 J671000 R103.00 SLH3.00
SM0..00 SM00.00 SM00z00 SM0z.00 SyuG000 U81..00 ZV11300 ZV11311 ZV4KC00 ZV57A00
ZV6D600 Z191.00 Z191100 Z191200 Z191211 Z191400 Z4B1.00 Z9KF400 Z9KF600 ZC22200
ZC2H.00 ZG23100 1D19.00 8CAM.00 {;
qui replace comorbid=7 if medcode=="`x'";
};
***HCV*************************************************************************
#delimit ;
foreach x in A704000 A705000 A707200 {;
qui replace comorbid=8 if medcode=="`x'";
};
***COCAINE*********************************************************************
#delimit ;
foreach x in 1T5.00 1T50.00 1T51.00 1T52.00 IT53.00 1T6.00 1T61.00 1T62.00
1T63.00 46QA.00 4I74.00 E242.00 E242000 E242100 E242200 E242Z00 E256.00 E256000
E256100 E256200 E256300 E256z00 Eu14.00 Eu14000 Eu14100 Eu14211 Eu14300 Eu14500
Eu1A.00 Eu1A000 Eu1A100 Eu1A200 R10B00 SL85000 T852000 R10B000 SL85000 T852000
TJ85000 U608312 ZR3Z.00 ZR3Z.11 {;
qui replace comorbid=9 if medcode=="`x'";
};
***HBV*************************************************************************
```

```
#delimit ;
foreach x in 43B4.00 43B5.00 7Q05200 A702.00 A703.00 A705100 A70700 A70710
Q409100 ZV02612 ZV02B00 {;
qui replace comorbid=10 if medcode=="`x'";
};
***CAD*****************************************************************
#delimit ;
foreach x in 792..11 SP07600 ZV45K00 ZV45K11 792..00 792..11 7920.00 7920.11
7920000 7920100 7920200 7920300 7920y00 7920z00 7921.00 7921.11 7921000 7921100
7921200 7921300 7921y00 7921z00 7922.00 7922.11 7922000 7922100 7922200 7922300
7922y00 7922z00 7923.00 7923.11 7923000 7923100 7923200 7923300 7923y00 7923z00
7924.00 7924000 7924100 7924200 7924300 7924400 7924y00 7924z00 7925.00 7925.11
7925000 7925100 7925200 7925300 7925400 7925y00 7925z00 7926.00 7926000 7926100
7926200 7926300 7926y00 7926z00 7927.00 7927500 7927y00 7927z00 792B.00 792B000
792B100 792By00 792Bz00 792C.00 792C000 792Cy00 792Cz00 792D.00 792Dy00 792Dz00
792y.00 792z.00 ZV45700 ZV45800 ZV45K00 ZV45K11 7928.00 7928.11 7928000 7928100
7928200 7928300 7928y00 7928z00 7929.00 7929000 7929100 7929111 7929200 7929300
7929400 7929500 7929600 7929y00 7929z00 792A.00 793G.00 793Gy00 793Gz00 ZV45L00
{;
qui replace comorbid=11 if medcode=="`x'";
};
***FAMILY HISTORY OF CAD***********************************************
#delimit ;
foreach x in 12C..00 12C2.00 12C3.00 12C5.11 12C5.12 12CI.00 ZV17300 ZV17311
ZV17312 ZV17400 {;
qui replace comorbid=12 if medcode=="`x'";
};
#delimit cr
tab comorbid
drop if comorbid==0
sort pracid patid bmi_date comorbid
by pracid patid bmi_date comorbid: gen litn=_n
**tab litn
keep if litn==1
drop medcode
reshape wide litn, i(pracid patid) j(comorbid)
sort pracid patid bmi_date
rename litn1 htn
rename litn2 dm
rename litn3 chol
rename litn4 renal
rename litn5 overweight
rename litn6 obese
rename litn7 alcohol
rename litn8 hcv
rename litn9 cocaine
rename litn10 hbv
rename litn11 cad
rename litn12 fxcad
foreach x in htn dm chol renal overweight obese alcohol hcv cocaine hbv cad
fxcad {
label var `x'
qui replace `x'=0 if `x'==.
}
compress
duplicates drop
move bmi_date htn
sort pracid patid bmi_date
save "/Users/kforde/Desktop/THIN/pat_comorbid_ever.dta", replace
```

```
***GENERATION OF MEDICATION FILE FROM THERAPY INFORMATION IN THIN BASED ON
PROJECT 2 COVARIATES***
clear
use "/Users/kforde/Desktop/THIN/Therapy_p10.dta"
merge m:1 pracid patid using  "/Users/kforde/Desktop/THIN/comorbid_set_up.dta"
keep if _merge==3
drop _merge
drop therflag bnf doscode prscqty prscdays private staffid prsctype opno
seqnoiss maxnoiss packsize dosgval locate drugsrce inprac therid cnsultid
modified sysdate
**drop if bmi_date< prscdate & prscdate!=.
drop if prscdate==.
*** observations dropped
drop prscdate
***MEDICATION EXPOSURES************************************************
***BETA BLOCKER SCRIPTS************************************************;
generate tx=0
#delimit ;
foreach x in 85788998 85789998 85844998 85959998 85960998 85961998 86017998
86051998 86098998 86099998 86782998 86783998 87288998 87290998 87881998
88077996 88077997 88077998 88137997 88137998 88160998 88161998 88362996
88362997 88362998 88388998 88405998 88406998 88521998 88890998 88896998
89029998 89288998 89290998 89396998 89397998 89656996 89656997 89656998
89658996 89658997 89658998 89659996 89659997 89659998 90918998 90919998
91079997 91079998 91220998 91326996 91326997 91326998 91478998 91554997
91554998 91766998 91837998 91884998 91887998 92151998 92809990 92810990
92827996 92827997 92827998 92828990 92829990 92830990 92889998 92890998
92926998 92930990 92931990 92932990 92933990 93076990 93077990 93149990
93209998 93210998 93219997 93219998 93295992 93305990 93306990 93342990
93344992 93345992 93561996 93561997 93561998 93562996 93562997 93562998
93563997 93563998 93608998 93625992 93691996 93691997 93691998 93692996
93692997 93692998 93746992 93746998 93771992 93798992 93822990 93824990
93825990 93835998 93850990 93851990 93852990 93877998 93885990 93886990
93905998 93922990 93934998 93935996 93935997 93935998 93937998 94156998
94179990 94180990 94181990 94182990 94257992 94258992 94269992 94272990
94273990 94278990 94279990 94280990 94302990 94303990 94304990 94305990
94306990 94307990 94319992 94320992 94321992 94421990 94422990 94422997
94422998 94423990 94424990 94467990 94468990 94493998 94497998 94498998
94499997 94499998 94545998 94549998 94550998 94551998 94566992 94567998
94573998 94623990 94624990 94633997 94633998 94634997 94634998 94635998
94637997 94637998 94668998 94669998 94670998 94671998 94672997 94672998
94679996 94679997 94679998 94681990 94682990 94682992 94726998 94732998
94783996 94783997 94783998 94802998 94804998 94808997 94808998 94813997
94813998 94922990 94923990 94924990 94928990 94929990 94929997 94929998
94930990 94931990 94953990 94954990 94955990 94957990 94958990 94959990
94960990 94961990 94962990 94963990 94964990 94965990 94966990 94967990
94973990 94974990 94975990 94976990 94983990 94984990 94985990 94986990
94988990 94989990 94990990 94991990 95011997 95011998 95012997 95012998
95014997 95014998 95021997 95021998 95022998 95120992 95140998 95159998
95161997 95161998 95192990 95253990 95254990 95258998 95264998 95265990
95267997 95267998 95316990 95317990 95376998 95377998 95378997 95378998
95389992 95421998 95464990 95467990 95468990 95513992 95514992 95514998
95515992 95515997 95515998 95516992 95517992 95518992 95519992 95520992
95522992 95566992 95596998 95602992 95605992 95616990 95617990 95660998
95661998 95662998 95680992 95695990 95765997 95765998 95766997 95766998
95824998 95825998 95826997 95826998 95855990 95956998 95957998 95995996
95995997 95995998 96001990 96002990 96003990 96004990 96017990 96018990
96025997 96025998 96031998 96032997 96032998 96035989 96035990 96043997
96043998 96064990 96065990 96070997 96070998 96083990 96093992 96205990
96237990 96238988 96238989 96238990 96243990 96276989 96276990 96289989
```

```
96289990 96292989 96292990 96294989 96294990 96319989 96319990 96324989
96324990 96332992 96337989 96337990 96346989 96346990 96348989 96348990
96349989 96349990 96465992 96473998 96585998 96630998 96682998 96683997
96683998 96710992 96711989 96711990 96719989 96719990 96732989 96732990
96735989 96735990 96770992 96782992 96828992 96843989 96843990 96849989
96849990 96856988 96856989 96856990 96864989 96864990 96895989 96895990
96907998 96924988 96924989 96924990 96940997 96940998 96971988 96971989
96971990 96990998 96992996 96992997 96992998 97098998 97099998 97100996
97100997 97100998 97101998 97102996 97102997 97102998 97103998 97104996
97104997 97104998 97105989 97105990 97122996 97122997 97122998 97123996
97123997 97123998 97124998 97125996 97125997 97125998 97137988 97137989
97137990 97164988 97164989 97164990 97168989 97168990 97333998 97350996
97350997 97350998 97365992 97465998 97481998 97558990 97587998 97588996
97588997 97588998 97589998 97590996 97590997 97590998 97597998 97680992
97691998 97710997 97710998 97727997 97727998 97740989 97740990 97745988
97745989 97745990 97769989 97769990 97786988 97786989 97786990 97797990
97801989 97801990 97816997 97816998 97861998 97890989 97890990 97943989
97943990 97949998 97968988 97968989 97968990 97982990 97983989 97983990
98039990 98041998 98055990 98093989 98093990 98094988 98094989 98094990
98117990 98143989 98143990 98145990 98146990 98179989 98179990 98181989
98181990 98224998 98284998 98285996 98285997 98285998 98286996 98286997
98286998 98298998 98299996 98299997 98299998 98300998 98301998 98344989
98344990 98361988 98361989 98361990 98367998 98368997 98368998 98380998
98381996 98381997 98381998 98398998 98404998 98446998 98447998 98448998
98450990 98451988 98451989 98451990 98482989 98482990 98483989 98483990
98591988 98591989 98591990 98600998 98603998 98604998 98636989 98636990
98650989 98650990 98655988 98655989 98655990 98676988 98676989 98676990
98702998 98775998 98798988 98798989 98798990 98799989 98799990 98816998
98851996 98851997 98851998 98894998 98937998 98961997 98961998 98963998
99031998 99033998 99067996 99067997 99067998 99068998 99088988 99088989
99088990 99089988 99089989 99089990 99090988 99090989 99090990 99118998
99132998 99151998 99169988 99169989 99169990 99173989 99173990 99277989
99277990 99278988 99278989 99278990 99322989 99322990 99323990 99417988
99417989 99417990 99418988 99418989 99418990 99419988 99419989 99419990
99459998 99460997 99460998 99461998 99500989 99500990 99501989 99501990
99536998 99538988 99538989 99538990 99539988 99539989 99539990 99705990
99706988 99706989 99706990 99707989 99707990 99708988 99708989 99708990
99709990 99710988 99710989 99710990 99711990 99712988 99712989 99712990
99808998 99819998 99891998 99892997 99892998 {;
qui replace tx=1 if drugcode=="`x'";
};
***CALCIUM CHANNEL BLOCKER SCRIPTS*****************************************;
#delimit ;
foreach x in 84189998 84299998 84300998 84649998 84734998 84888998 84889998
84901998 84922998 84924998 84925998 85014998 85015998 85016998 85017998
85018998 85019998 85025998 85395998 85396998 85775998 85791998 85792998
85850998 85900998 86031998 86052998 86053998 86054998 86107998 86108998
86124998 86125998 86137998 86583998 86760998 86929998 86930998 86931998
86973998 86974998 86989998 86990998 87016998 87294998 87295998 87297998
87298998 87446998 87447998 87507998 87508998 87509998 87510998 87511998
87512998 87538998 87539998 87572998 87573998 87653998 87654998 87879998
87888998 87889998 87890998 87897998 87898998 87963998 87969998 87984998
87992997 87992998 88034998 88055998 88062998 88083997 88083998 88116998
88234997 88234998 88290997 88290998 88319996 88319997 88319998 88328998
88410998 88418997 88418998 88425998 88433998 88441998 88448997 88448998
88457998 88461998 88491998 88541998 88551997 88551998 88837996 88837997
88837998 88877997 88877998 88884998 88945996 88945997 88945998 89020998
89024998 89057997 89057998 89067998 89085997 89085998 89087996 89087997
```

```
89087998 89103998 89145998 89186998 89190997 89190998 89459998 89519996
89519997 89519998 89522996 89522997 89522998 89618998 89652997 89652998
89704997 89704998 89768997 89768998 90181998 90182998 90210998 90294998
90392998 90433998 90434998 90445997 90445998 90559998 90629996 90629997
90629998 90706996 90706997 90706998 90722996 90722997 90722998 90781998
90854998 90871997 90871998 90981998 91003998 91018998 91085998 91145997
91145998 91168997 91168998 91270998 91344998 91358996 91358997 91358998
91372998 91422997 91422998 91518998 91565997 91565998 91780997 91780998
91784998 91796998 91810998 92042997 92042998 92202998 92204997 92204998
92341996 92341997 92341998 92354998 92355998 92527996 92527997 92527998
92588998 92627996 92627997 92627998 92727997 92727998 92753996 92753997
92753998 92754996 92754997 92754998 92762997 92762998 92824996 92824997
92824998 92826998 92835990 92846990 92849996 92849997 92849998 92968997
92968998 92969997 92969998 93002992 93013992 93014996 93014997 93014998
93133998 93232997 93232998 93247992 93248992 93249992 93251997 93251998
93332990 93392990 93396990 93523997 93523998 93524997 93524998 93599996
93599997 93599998 93600996 93600997 93600998 93710997 93710998 93767992
93772998 93776990 93929998 94050992 94055997 94055998 94082998 94111998
94118998 94145996 94145997 94145998 94213990 94214990 94275992 94319990
94320990 94340990 94341990 94376990 94383990 94384990 94386992 94465998
94474998 94475998 94520996 94520997 94520998 94658990 94659990 94668998
94669998 94714998 94715998 94716998 94726998 94732998 94739998 94740997
94740998 94741997 94741998 94742998 94748990 94749990 94766996 94766997
94766998 94769990 94770990 94774996 94774997 94774998 94775996 94775997
94775998 94785998 94786998 94792990 94793990 94810998 94857990 94858990
94860998 94869990 94875990 94876990 94924992 94979990 94980990 94981990
94982990 94994990 94995990 95010990 95011990 95208990 95209990 95328990
95329990 95355990 95356990 95377990 95378990 95407992 95409998 95465990
95572992 95588990 95589990 95626997 95626998 95628996 95628997 95628998
95631996 95631997 95631998 95652990 95653990 95714990 95715990 95716990
95723998 95724996 95724997 95724998 95730997 95730998 95836997 95836998
95852990 95853990 96016990 96098992 96164990 96168990 96169990 96365989
96365990 96368996 96368997 96368998 96416989 96416990 96505992 96640989
96640990 96684990 96822990 96831990 96832988 96832989 96832990 96857992
96870990 96876989 96876990 96882992 96904988 96904989 96904990 96908997
96908998 96927992 96992989 96992990 96993988 96993989 96993990 97048992
97049992 97090988 97090989 97090990 97106989 97106990 97116989 97116990
97122990 97159989 97159990 97189988 97189989 97189990 97241998 97272997
97272998 97382996 97382997 97382998 97481998 97510998 97635998 97707996
97707997 97707998 97720996 97720997 97720998 97733989 97733990 97733996
97733997 97733998 97764988 97764989 97764990 97790998 97821998 97822998
97895989 97895990 97896988 97896989 97896990 97939988 97939989 97939990
98019990 98092990 98231992 98267998 98290997 98290998 98338997 98338998
98348990 98366988 98366989 98366990 98367989 98367990 98383990 98473989
98473990 98565990 98567988 98567989 98567990 98624998 98625996 98625997
98625998 98642988 98642989 98642990 98755998 98756998 98886997 98886998
98888997 98888998 98985996 98985997 98985998 99013997 99013998 99040989
99040990 99086998 99334988 99334989 99334990 99335988 99335989 99335990
99336988 99336989 99336990 99407998 99458998 99620988 99620989 99620990
99621990 99622990 99721988 99721989 99721990 99722989 99722990 99723989
99723990 99724989 99724990 99809998 99810997 99810998 {;
qui replace tx=2 if drugcode=="`x'";
};
***DIURETIC SCRIPTS************************************************************;
#delimit ;
foreach x in 84035998 84176998 84177998 84178998 84253998 85218998 85219998
86039998 86040998 86041998 86048998 86089998 86090998 86092998 86093998
86128998 86520998 86521998 86522998 87212998 87213998 87214998 87216998
```

87217998 87218998 87219998 87288998 87421998 87422998 87424998 87426998
87427998 87428998 87515998 87516998 87955998 87956998 88154998 88955998
89061997 89061998 89237998 89288998 89290998 89292997 89292998 89305998
89320998 89622998 90026998 90077998 90392998 90433998 90434998 90547998
90549998 90740997 90740998 91079997 91079998 91089997 91089998 91211998
91240998 91577998 91607998 91818998 92670998 92736997 92736998 92797996
92797997 92797998 92803997 92803998 92820997 92820998 92828996 92828997
92828998 92976990 93101998 93102998 93192998 93194998 93195998 93209998
93210998 93211997 93211998 93212996 93212997 93212998 93219997 93219998
93341992 93351992 93394992 93399992 93399998 93401998 93605998 93612998
93633998 93661992 93662992 93690992 93734992 93735992 93748997 93748998
93752992 93773992 93831990 93832990 93846998 93847992 93891992 93922990
94003992 94004992 94024992 94068990 94100992 94101992 94102992 94103992
94104992 94105992 94106992 94146996 94146997 94146998 94243992 94266990
94389990 94390990 94419992 94422997 94422998 94429990 94430990 94434992
94442990 94467990 94468990 94501992 94537992 94541992 94546996 94546997
94546998 94547996 94547997 94547998 94549998 94550998 94551998 94553997
94553998 94591990 94592990 94593990 94605992 94617998 94625990 94633997
94633998 94634997 94634998 94635998 94640990 94656990 94657990 94663990
94670998 94671998 94675998 94701992 94702992 94727990 94729990 94747990
94765997 94765998 94768990 94802992 94803992 94804992 94811992 94812992
94813992 94814992 94815992 94816992 94817992 94818992 94819992 94820992
94823992 94825992 94839992 94865998 94868997 94868998 94869997 94869998
94889990 94903998 94917998 94949990 94950990 94951990 94952990 94991992
94992990 94993990 95010998 95011997 95011998 95012997 95012998 95014997
95014998 95021997 95021998 95123997 95123998 95124998 95125998 95126998
95128998 95129998 95159998 95161997 95161998 95257998 95258990 95260997
95260998 95261998 95273992 95320990 95321990 95335998 95336998 95399990
95407990 95452990 95453990 95469998 95477990 95534990 95535990 95540992
95596998 95616990 95616998 95617990 95635990 95636990 95641990 95695990
95765997 95765998 95827998 95861998 95862998 95897998 95908998 95974998
96013990 96062990 96069998 96070997 96070998 96129997 96129998 96202997
96202998 96203998 96205998 96206998 96207998 96208998 96209997 96209998
96210998 96211996 96211997 96211998 96212997 96212998 96218990 96219990
96222998 96313989 96313990 96379990 96380990 96434990 96439990 96474998
96475996 96475997 96475998 96585998 96586998 96587998 96682998 96683997
96683998 96684998 96685997 96685998 96696998 96794990 96862990 96862992
96883992 96886990 96888990 96889998 96890998 96895989 96895990 96899990
96912990 96913990 96934990 96967990 96976990 96990998 97060998 97086990
97126989 97126990 97128992 97168989 97168990 97188997 97188998 97189996
97189997 97189998 97196998 97201996 97201997 97201998 97217997 97217998
97224996 97224997 97224998 97232990 97251998 97279996 97279997 97279998
97295998 97425998 97486998 97492992 97556990 97557990 97600998 97636997
97636998 97637996 97637997 97637998 97710997 97710998 97721998 97731998
97747990 97769989 97769990 97771990 97804990 97872989 97872990 97877989
97877990 97890989 97890990 97892990 97921992 97949998 97988990 98009988
98009989 98009990 98082990 98083988 98083989 98083990 98095989 98095990
98096988 98096989 98096990 98098990 98144989 98144990 98179989 98179990
98180988 98180989 98180990 98181989 98181990 98247989 98247990 98301998
98336989 98336990 98344989 98344990 98367998 98398998 98411990 98412990
98446998 98447998 98515990 98516990 98600998 98616988 98616989 98616990
98636989 98636990 98641989 98641990 98731997 98731998 98736996 98736997
98736998 98742996 98742997 98742998 98775998 98797989 98797990 98811989
98811990 98816998 98943997 98943998 99031998 99068998 99069998 99151998
99161998 99162997 99162998 99266998 99272998 99328997 99328998 99366998
99368998 99372998 99373998 99378988 99378989 99378990 99379988 99379989
99379990 99380988 99380989 99380990 99405997 99405998 99424998 99429998
99459998 99493998 99547998 99548997 99548998 99620998 99677997 99677998
99679990 99680990 99686998 99707998 99708998 99709998 99734998 99790989
99790990 99791989 99791990 99792989 99792990 99808998 99819998 99849992

```
99853992 99874990 99875990 99876990 99877990 99878990 99894997 99894998
99916998 99943997 99943998 {;
qui replace tx=3 if drugcode=="`x'";
};
***VASODILATOR ANTI-HYPERTENSIVE SCRIPTS************************************;
#delimit ;
foreach x in 86826998 93299992 93699992 94182992 94186992 94493992 94652992
94653992 95112992 95114992 95275997 95275998 95276997 95276998 95799996
95799997 95799998 95800996 95800997 95800998 96214998 96638992 96739990
96863989 96863990 97117989 97117990 97214997 97214998 98617990 98660989
98660990 99185989 99185990 99186989 99186990 99568989 99568990 99569989
99569990 99570989 99570990 99669998 99944996 99944997 99944998 {;
qui replace tx=4 if drugcode=="`x'";
};
***CENTRAL ANTI-HYPERTENSIVE SCRIPTS*****************************************;
#delimit ;
foreach x in 90609996 90609997 90609998 90613996 90613997 90613998 91268998
92094998 93204998 93397992 93580998 93581998 93659998 93773992 93867992
94204990 94205990 94206990 94332992 94402992 94501992 94536990 94537990
94537992 94538990 94541992 94547990 94548990 94549990 94553990 94554990
94555990 94556990 94557990 94571990 94577990 94578990 94579990 94580990
94581990 94612990 94614990 94616990 94628990 94629990 94630990 94756990
94757990 94758990 94902992 94958992 95015998 95088992 95236990 95238990
95273992 95276992 95296990 95297990 95323998 95324998 95325997 95325998
95329998 95330990 95330998 95331998 95460990 95540992 95853998 95855998
95856997 95856998 95870998 95871997 95871998 95975998 96323992 96631998
96632997 96632998 96633998 96848988 96848989 96848990 97109989 97109990
97190996 97190997 97190998 97204996 97204997 97204998 97337998 97551998
97552998 97741992 98276998 98292996 98292997 98292998 98406998 98454998
98488989 98488990 98488998 98489989 98489990 98665988 98665989 98665990
99102997 99102998 99192998 99194998 99506988 99506989 99506990 99507988
99507989 99507990 99508988 99508989 99508990 99509988 99509989 99509990
99733998 99783997 99783998 99850997 99850998 99975996 99975997 99975998
99996998 94273992 95399992 {;
qui replace tx=5 if drugcode=="`x'";
};
***ADRENERGIC NEURONE ANTI-HYPERTENSIVE SCRIPTS******************************;
#delimit ;
foreach x in 87092998 93199997 93199998 93694998 94178992 94372992 95078992
95079992 95085992 95675992 95958997 95958998 95959997 95959998 96456997
96456998 97129990 97942990 98204990 99526996 99526997 99526998 99678997
99678998 99782997 99782998 {;
qui replace tx=6 if drugcode=="`x'";
};
***ALPHA ADRENERGIC BLOCKING ANTI-HYPERTENSIVE SCRIPTS***********************;
#delimit ;
foreach x in 84266998 84444998 84445998 84448998 84463998 84464998 84465998
84534998 84548998 84832998 85046998 85375998 86348998 88481997 88481998
88958997 88958998 89337998 89431998 90938998 90939998 93144998 93482990
93483990 93484990 93535998 93536998 93703998 93757990 93758990 93759990
93919990 93920990 93921990 94137992 94159998 94183990 94184990 94192990
94193990 94194990 94263990 94264990 94265990 94375990 94447992 94522992
94541996 94541997 94541998 94542996 94542997 94542998 94825998 94826996
94826997 94826998 94888998 94889996 94889997 94889998 94890998 94891996
94891997 94891998 94916992 95306990 95307990 95308990 95417990 95418998
95419996 95419997 95419998 95422992 95423990 95423992 95424990 95425990
95521990 95522990 95523990 95525990 95526990 95530990 95542998 95543998
95545998 95546990 95547990 95548990 95612990 95613990 95614990 95623990
95624990 95625990 95642990 95643990 95644990 95647990 95648990 95649990
```

```
95736990 95737990 95738990 95881990 95882990 95922990 95923990 95924990
95938990 95940990 95942990 95958990 95959990 95960990 95961990 95962990
95963990 95975990 95989990 95990990 95991990 95998990 95999990 96000990
96053990 96054990 96125997 96125998 96225990 96226990 96227990 96804998
97267998 97727990 97728988 97728989 97728990 97875990 97904990 98358990
98359988 98359989 98359990 98457988 98457989 98457990 98458990 98459988
98459989 98459990 99079996 99079997 99079998 99095998 99135996 99135997
99279998 99545998 99546996 99546997 99546998 99760998 99918997 99918998
84645998 84998998 85439998 85764998 85978998 86086998 86087998 86105998
86106998 86115998 86121998 86122998 86123998 86738998 86739998 88985998
89403998 90373998 90374998 94445998 94446998 94859990  99135998 {;
qui replace tx=7 if drugcode=="`x'";
};
***ANGIOTENSIN CONVERTING ENZYMES INHIBITOR SCRIPTS************************;
#delimit ;
foreach x in 84314998 84315998 84572998 85460998 85496998 86129998 86138998
86350998 86454998 86895998 86896998 86897998 86898998 87064998 87223998
87224998 87225998 87226998 87261998 87370998 87371998 87372998 87515998
87516998 87621998 87622998 87699998 87700998 87701998 87702998 87812998
87899998 87900998 87901998 87902998 87903998 87904998 87905998 87906998
88349996 88349997 88349998 88350998 88457998 88461998 88508996 88508997
88508998 88714998 88955998 89058996 89058997 89058998 89061997 89061998
89069996 89069997 89069998 89102996 89102997 89102998 89305998 89424998
89543998 90049998 90410996 90410997 90410998 90445997 90445998 90533996
90533997 90533998 90576996 90576997 90576998 90740997 90740998 90816997
90816998 90817997 90817998 91065998 91444998 91780997 91780998 91958996
91958997 91958998 92731998 92732998 92887996 92887997 92887998 92959990
92960990 92961990 93006990 93007990 93008990 93032990 93033990 93034990
93043990 93044990 93045990 93046990 93101998 93102998 93136997 93136998
93137997 93137998 93329990 93330990 93331990 93335990 93336990 93337990
93343990 93344990 93393990 93394990 93459996 93459997 93459998 93567996
93567997 93567998 93568996 93568997 93568998 93597997 93597998 93598997
93598998 93727992 93748997 93748998 93797990 93798990 93883990 93884990
93895990 93896990 93897990 93898990 93915990 93916990 93917990 93918990
93928996 93928997 93928998 93931990 93932990 93974990 93975990 93979990
93980990 94014990 94015990 94040990 94040992 94041990 94139998 94200990
94252990 94253990 94254990 94255990 94266990 94269990 94270990 94271990
94315990 94316990 94317990 94318990 94330990 94331990 94332990 94333990
94336990 94337990 94338990 94339990 94359990 94360990 94361990 94362990
94395996 94395997 94395998 94396996 94396997 94396998 94409990 94410990
94411990 94412990 94509990 94510990 94511990 94512990 94515990 94521990
94522990 94523990 94524990 94553997 94553998 94560990 94561990 94561992
94562990 94563990 94572990 94573990 94574990 94575990 94603996 94603997
94603998 94633990 94634990 94635990 94636990 94649990 94650990 94651990
94652990 94656990 94657990 94713990 94717990 94719990 94722990 94727990
94729990 94748998 94749996 94749997 94749998 94750998 94751996 94751997
94751998 94752998 94753996 94753997 94753998 94826990 94827990 94828990
94833990 94834990 94835990 94860990 94861990 94862990 94863990 94867996
94867997 94867998 94868997 94868998 94869997 94869998 94896990 94897990
94898990 94899990 94909990 94910990 94911990 94912990 94932990 94933990
94934990 94935990 94939990 94940990 94941990 94942990 94943990 94944990
94945990 94948990 94949990 94950990 94951990 94952990 94972990 94992990
94993990 95042990 95043990 95044990 95045990 95054990 95055990 95056990
95057990 95058990 95059990 95060990 95061990 95071990 95072990 95073990
95074990 95096990 95097990 95098990 95099990 95100990 95101990 95102990
95103990 95127990 95128990 95129990 95133990 95134990 95135990 95136990
95137990 95138990 95139990 95144990 95145990 95146990 95147990 95148990
95149990 95150990 95151990 95199990 95249990 95250990 95318990 95319990
95383990 95384990 95385990 95390990 95391990 95392990 95393990 95399990
95407990 95477990 95635990 95636990 95641990 95673990 95674990 95675990
```

95676990 95678990 95679990 95680990 95681990 95682990 95683990 95684990
95685990 95688990 95689990 95690990 95691990 95699990 95700990 95701990
95702990 95712990 95713990 95717990 95721990 95722990 95723990 95724990
95787990 95788990 95789990 95790990 96099992 96178990 96179990 96180990
96181990 96194990 96195990 96196990 96273990 96274988 96274989 96274990
96300990 96301988 96301989 96301990 96305988 96305989 96305990 96562990
96563988 96563989 96563990 96646988 96646989 96646990 96657990 96658988
96658989 96658990 96660990 96661988 96661989 96661990 96664990 96665988
96665989 96665990 96667990 96668988 96668989 96668990 96678990 96679988
96679989 96679990 96686990 96687988 96687989 96687990 96689990 96690988
96690989 96690990 96795996 96795997 96795998 96796992 96796996 96796997
96796998 96886990 96888996 96888997 96888998 96900990 96902988 96902989
96902990 96968988 96968989 96968990 97003988 97003989 97003990 97039990
97040990 97041990 97060998 97086992 97139988 97139989 97139990 97347992
97359997 97359998 97491998 97793988 97793989 97793990 97794988 97794989
97794990 97819988 97819989 97819990 97820988 97820989 97820990 97821988
97821989 97821990 97831988 97831989 97831990 97832988 97832989 97832990
97833988 97833989 97833990 98058990 98153998 98169997 98169998 98216996
98216997 98216998 98471997 98471998 98877996 98877997 98877998 99162997
99162998 99266998 99272998 99328997 99328998 99384996 99384997 99384998
99660996 99660997 99660998 99851996 99851997 99851998 84035998 84036998
84037998 84038998 84039998 84040998 84041998 84061998 92799990 92800990
92801990 92802990 92817990 92842990 92843990 92844990 92847990 92848990
92850990 92852990 92856990 92857990 92858990 92885990 92886990 92887990
84253998 84314998 84315998 84572998 85460998 85496998 86129998 86138998
86350998 86454998 86895998 86896998 86897998 86898998 87064998 87223998
87224998 87225998 87226998 87261998 87370998 87371998 87372998 87515998
87516998 87621998 87622998 87699998 87700998 87701998 87702998 87812998
87899998 87900998 87901998 87902998 87903998 87904998 87905998 87906998
88349996 88349997 88349998 88350998 88457998 88461998 88508996 88508997
88508998 88714998 88955998 89058996 89058997 89058998 89061997 89061998
89069996 89069997 89069998 89102996 89102997 89102998 89305998 89424998
89543998 90049998 90410996 90410997 90410998 90445997 90445998 90533996
90533997 90533998 90576996 90576997 90576998 90740997 90740998 90816997
90816998 90817997 90817998 91065998 91444998 91780997 91780998 91958996
91958997 91958998 92731998 92732998 92887996 92887997 92887998 92959990
92960990 92961990 93006990 93007990 93008990 93032990 93033990 93034990
93043990 93044990 93045990 93046990 93101998 93102998 93136997 93136998
93137997 93137998 93329990 93330990 93331990 93335990 93336990 93337990
93343990 93344990 93393990 93394990 93459996 93459997 93459998 93567996
93567997 93567998 93568996 93568997 93568998 93597997 93597998 93598997
93598998 93727992 93748997 93748998 93797990 93798990 93883990 93884990
93895990 93896990 93897990 93898990 93915990 93916990 93917990 93918990
93928996 93928997 93928998 93931990 93932990 93974990 93975990 93979990
93980990 94014990 94015990 94040990 94040992 94041990 94139998 94200990
94252990 94253990 94254990 94255990 94266990 94269990 94270990 94271990
94315990 94316990 94317990 94318990 94330990 94331990 94332990 94333990
94336990 94337990 94338990 94339990 94359990 94360990 94361990 94362990
94395996 94395997 94395998 94396996 94396997 94396998 94409990 94410990
94411990 94412990 94509990 94510990 94511990 94512990 94515990 94521990
94522990 94523990 94524990 94553997 94553998 94560990 94561990 94561992
94562990 94563990 94572990 94573990 94574990 94575990 94603996 94603997
94603998 94633990 94634990 94635990 94636990 94649990 94650990 94651990
94652990 94656990 94657990 94713990 94717990 94719990 94722990 94727990
94729990 94748998 94749996 94749997 94749998 94750998 94751996 94751997
94751998 94752998 94753996 94753997 94753998 94826990 94827990 94828990
94833990 94834990 94835990 94860990 94861990 94862990 94863990 94867996
94867997 94867998 94868997 94868998 94869997 94869998 94896990 94897990
94898990 94899990 94909990 94910990 94911990 94912990 94932990 94933990
94934990 94935990 94939990 94940990 94941990 94942990 94943990 94944990

```
94945990 94948990 94949990 94950990 94951990 94952990 94972990 94992990
94993990 95042990 95043990 95044990 95045990 95054990 95055990 95056990
95057990 95058990 95059990 95060990 95061990 95071990 95072990 95073990
95074990 95096990 95097990 95098990 95099990 95100990 95101990 95102990
95103990 95127990 95128990 95129990 95133990 95134990 95135990 95136990
95137990 95138990 95139990 95144990 95145990 95146990 95147990 95148990
95149990 95150990 95151990 95199990 95249990 95250990 95318990 95319990
95383990 95384990 95385990 95390990 95391990 95392990 95393990 95399990
95407990 95477990 95635990 95636990 95641990 95673990 95674990 95675990
95676990 95678990 95679990 95680990 95681990 95682990 95683990 95684990
95685990 95688990 95689990 95690990 95691990 95699990 95700990 95701990
95702990 95712990 95713990 95717990 95721990 95722990 95723990 95724990
95787990 95788990 95789990 95790990 96099992 96178990 96179990 96180990
96181990 96194990 96195990 96196990 96273990 96274988 96274989 96274990
96300990 96301988 96301989 96301990 96305988 96305989 96305990 96562990
96563988 96563989 96563990 96646988 96646989 96646990 96657990 96658988
96658989 96658990 96660990 96661988 96661989 96661990 96664990 96665988
96665989 96665990 96667990 96668988 96668989 96668990 96678990 96679988
96679989 96679990 96686990 96687988 96687989 96687990 96689990 96690988
96690989 96690990 96795996 96795997 96795998 96796992 96796996 96796997
96796998 96886990 96888996 96888997 96888998 96900990 96902988 96902989
96902990 96968988 96968989 96968990 97003988 97003989 97003990 97039990
97040990 97041990 97060998 97086992 97139988 97139989 97139990 97347992
97359997 97359998 97491998 97793988 97793989 97793990 97794988 97794989
97794990 97819988 97819989 97819990 97820988 97820989 97820990 97821988
97821989 97821990 97831988 97831989 97831990 97832988 97832989 97832990
97833988 97833989 97833990 98058990 98153998 98169997 98169998 98216996
98216997 98216998 98471997 98471998 98877996 98877997 98877998 99162997
99162998 99266998 99272998 99328997 99328998 99384996 99384997 99384998
99660996 99660997 99660998 99851996 99851997 99851998 84035998 84036998
84037998 84038998 84039998 84040998 84041998 84061998 92799990 92800990
92801990 92802990 92817990 92842990 92843990 92844990 92847990 92848990
92850990 92852990 92856990 92857990 92858990 92885990 92886990 92887990
84253998 88425998 88433998 88441998 {;
qui replace tx=8 if drugcode=="`x'";
};
***ANGIOTENSIN II RECEPTOR ANTAGONIST SCRIPTS*********************************;
#delimit ;
foreach x in 84168998 84169998 84176998 84177998 84178998 84313998 84922998
84924998 84925998 85014998 85015998 85016998 85017998 85018998 85019998
85218998 85219998 86039998 86040998 86089998 86090998 86092998 86093998
86520998 86521998 86522998 87027998 87028998 87152998 87153998 87212998
87213998 87214998 87421998 87422998 87424998 87426998 87427998 87428998
88335998 88506996 88506997 88506998 88507996 88507997 88507998 89282996
89282997 89282998 89283996 89283997 89283998 89292997 89292998 89513996
89513997 89513998 89514996 89514997 89514998 90077998 90432996 90432997
90432998 90503996 90503997 90503998 90504998 90547998 90549998 90789998
91089997 91089998 91240998 91520998 91571996 91571997 91571998 91577998
91861998 91867998 91868998 91872998 91873998 92516997 92516998 92517997
92517998 92974990 92990996 92990997 92990998 97251998 97649998 99686998 {;
qui replace tx=9 if drugcode=="`x'";
};
***RENIN INHIBITOR SCRIPTS****************************************************;
#delimit ;
foreach x in 84429998 84430998 84431998 84432998 {;
qui replace tx=10 if drugcode=="`x'";
};
***GANGLION BLOCKER SCRIPTS***************************************************;
#delimit ;
foreach x in 94232992 95112998 95113998 95252992 97900990 {;
```

```
qui replace tx=11 if drugcode=="`x'";
};
***INSULIN SCRIPTS*************************************************************;
#delimit ;
foreach x in 96044992 84421998 85591998 86044998 86045998 86046998 86047998
86174998 86176998 86184998 86214998 86215998 86236998 86237998 86251998
86252998 86253998 86254998 86255998 86256998 86263998 86264998 86265998
86314998 86549998 86551998 86553998 88003998 88413998 88851998 88999998
90012998 90015998 90379998 90689998 90690998 90691998 91274998 91509998
91612998 93467992 94202992 94292998 94477992 94948998 95158992 95162992
96047998 96048998 96049998 96050998 96063998 96064998 96065998 96281992
96286992 96290992 96295992 96688992 96787992 97322997 97322998 97524998
97525998 97602992 98198998 98227998 98474990 98480998 98507998 98982998
99356998 99402998 99553998 99557998 99976992 84422998 84779998 86028998
86029998 86077998 86078998 86080998 86081998 86168998 86169998 86177998
86178998 86180998 86186998 86187998 86188998 86189998 86190998 86191998
86193998 86194998 86238998 86239998 86240998 86241998 86242998 86243998
86245998 86246998 86247998 86248998 86249998 86250998 86259998 86260998
86261998 86262998 86266998 86267998 86268998 86269998 86270998 86271998
86272998 86274998 86275998 86276998 86278998 86279998 86280998 86284998
86286998 86287998 86291998 86294998 86295998 86298998 86300998 86301998
86303998 86304998 86305998 86306998 86308998 86309998 86310998 86311998
87471998 87472998 87473998 87967997 87967998 88978998 88995998 89554998
89555998 89888998 89990997 89990998 90168998 90169998 90681996 90681997
90681998 90682996 90682997 90682998 90683997 90683998 90684996 90684997
90684998 90685998 90686998 90687998 90688998 90697996 90697997 90697998
90698998 91160998 91273997 91273998 91275996 91275997 91275998 91276998
91289998 91290996 91290997 91290998 91291997 91291998 91292996 91292997
91292998 91293997 91293998 91294997 91294998 91295998 91505998 91700998
91701998 91758998 92323998 92376996 92376997 92376998 92555998 92906998
92907998 92908998 92909998 92932998 93137992 93139992 94201992 94296998
94297998 94298998 94299998 94319998 94322998 94328998 94337998 94413998
94436998 95163992 95164992 95165992 95168992 95846992 96045998 96046992
96046998 96051998 96052998 96053996 96053997 96053998 96054998 96055998
96056998 96057998 96058998 96059998 96060998 96061998 96062998 96064992
96076992 96282992 96283992 96284992 96285992 96287992 96289992 96291992
96292992 96293992 96294992 96548992 96689992 96792992 96794992 96795992
97051997 97051998 97052996 97052997 97052998 97053998 97244992 97323998
97526998 97527998 97528998 97598992 97599992 97600992 97601992 97639992
97854998 98048990 98225998 98226998 98228996 98228997 98228998 98268998
98481997 98481998 98505998 98506998 98525990 98817998 98895998 99144998
99196998 99359998 99360998 99401998 99415998 99480998 99532998 99533998
99554998 99556998 99977992 99978992 81307994 81423994 81468994 82463994
82464994 82465994 83542994 83543994 83544994 83548994 83549994 83551994
83993994 85557994 85558994 85559994 85560994 87008994 87317994 87319994
87320994 87321994 87322994 87410994 87411994 87412994 87415994 87416994
88210994 88211994 88973994 88974994 89081994 89082994 89662994 89663994
90098994 90508994 90817994 90818994 90819994 90820994 90821994 90828994
90829994 90830994 80085994 86173998 86182998 86183998 86185998 86312998
86313998 86315998 86316998 86317998 86319998 {;
qui replace tx=12 if drugcode=="`x'";
};
***SULPHONYLUREA SCRIPTS********************************************************;
#delimit ;
foreach x in 85901998 86018998 88135998 88334998 88355998 88447996 88447997
88447998 88449996 88449997 88449998 91247998 91407998 91559998 92518997
92518998 93093990 93094990 93095990 93096990 93118990 93119990 93120990
93121990 93125990 93126990 93127990 93128990 93322990 93323990 93324990
93370990 93371990 93372990 93373990 93542990 93543990 93544990 93545990
```

```
93561990 93562990 93563990 93564990 93781990 93867990 93901990 94215992
94333992 94371992 94470992 95025990 95148998 95149997 95149998 95150997
95150998 95255992 95256992 95288990 95403990 95422990 95446990 95601990
95672992 95674992 95870992 95898990 96220990 96221990 96264998 96280998
96281998 96282997 96282998 96283997 96283998 96427990 96495990 96559990
96615989 96615990 96687998 96699990 96707990 96755997 96755998 96795990
96893990 96981998 97026990 97032990 97057997 97057998 97089998 97097997
97097998 97109998 97127997 97127998 97133992 97146990 97154990 97158990
97166990 97202990 97236992 97303998 97537997 97537998 97538990 97552989
97552990 97583997 97583998 97590990 97717998 97751989 97751990 97775989
97775990 97834990 97889990 97938990 98053990 98133990 98188989 98188990
98418989 98418990 98548990 98643989 98643990 98664989 98664990 99041990
99145998 99195998 99230998 99246989 99246990 99247989 99247990 99347990
99348990 99349990 99419998 99580989 99580990 99581989 99581990 99582989
99582990 99587998 99588998 99589998 99591998 99668997 99668998 99754998
99764997 99764998 99787998 83836998 83837998 83838998 83839998 83916998
83949998 92831990 {;
qui replace tx=13 if drugcode=="`x'";
};
***BIGUANIDE SCRIPTS************************************************************;
#delimit ;
foreach x in 85673998 85674998 87053998 87054998 87536998 87882998 87883998
89155997 89155998 91221997 91221998 93167990 94235992 94246990 94248990
94280992 94473990 94474990 94518990 94519990 94978990 95076992 95228990
95239990 95270992 95271992 95272992 95298990 95299990 95380990 95381990
95413992 95414992 95599990 95600990 95880997 95880998 96110990 96111990
96270989 96270990 96296989 96296990 96850989 96850990 97087997 97087998
97110989 97110990 98125989 98125990 98493989 98493990 98494990 98654989
98654990 99149989 99149990 99513989 99513990 99514989 99514990 99590997
99590998 84008998 84009998 84010998 84011998 85622998 85624998 85625998
87165998 87166998 87179998 87180998 87181998 87182998 87770998 87771998
87772998 87773998 87774998 87775998 {;
qui replace tx=14 if drugcode=="`x'";
};
***OTHER_DM SCRIPTS*************************************************************;
#delimit ;
foreach x in 84639998 84640998 85266998 85267998 85268998 85622998 85624998
85625998 87165998 87166998 87179998 87180998 87181998 87182998 87770998
87771998 87772998 87773998 87774998 87775998 87884998 87885998 88131996
88131997 88131998 88132996 88132997 88132998 88523996 88523997 88523998
88528996 88528997 88528998 89763996 89763997 89763998 90048996 90048997
90048998 91923996 91923997 91923998 91924996 91924997 91924998 92237997
92237998 92238997 92238998 95084992 96051992 96251998 96252998 96253996
96253997 96253998 97899998 98475997 98475998 98803998 98915997 98915998
84338998 84341998 84008998 84009998 84010998 84011998 {;
qui replace tx=15 if drugcode=="`x'";
};
***STATIN SCRIPTS***************************************************************;
#delimit ;
foreach x in 86020998 86467998 86468998 86787998 86788998 86789998 87373998
87416998 87417998 87418998 87916998 87917998 87918998 88534998 89153996
89153997 89153998 89154996 89154997 89154998 89306996 89306997 89306998
89311996 89311997 89311998 90309998 90310998 90973998 91194998 92220998
92408998 92409998 92410998 92447997 92447998 92448997 92448998 92471998
92539998 92804996 92804997 92804998 92805997 92805998 93230990 93231990
93232990 93243996 93243997 93243998 93244996 93244997 93244998 93504990
93619996 93619997 93619998 93620996 93620997 93620998 93870990 93871990
93872990 93873990 94196990 94197990 94198990 94199990 94321990 94322990
94323990 94324990 94325990 94326990 94327990 94328990 94329990 94363990
```

```
94364990 94365990 94392990 94393990 94394990 94406990 94407990 94408990
94702990 94703990 94704990 94777990 94778990 94779990 94781990 94782990
94783990 94787990 94788990 94789990 94794990 94795990 94796990 94802990
94803990 94804990 94805990 94806990 94807990 94811990 94812990 94813990
94829990 94830990 94831990 94849990 94850990 94851990 94919990 94920990
94921990 94927990 95185990 95277990 95278990 95279990 95372990 95373990
95374990 95405990 95406990 95408990 95414990 95415990 95416990 95442990
95443990 95444990 95445990 95448990 95449990 95450990 95451990 95471990
95472990 95473990 95474990 95475990 95476990 95478990 95479990 95480990
95481990 95482990 95483990 95486990 95487990 95488990 95493990 95494990
95495990 95500990 95501990 95502990 95508990 95549990 95550990 95551990
86791998 86794998
86795998 86796998 86797998 86798998 {;
qui replace tx=16 if drugcode=="`x'";
};
***ASA SCRIPTS*****************************************************************;
#delimit ;
foreach x in 85086998 86487998 87935998 87936998 88050998 88136998 88478998
88496998 88498998 88500998 88820998 89217998 89218998 89625998 89662998
89682998 89740998 89787998 89898998 90078996 90078997 90078998 90140998
90143998 90202998 90204998 90223998 90224998 90277998 90278998 90377997
90377998 90378998 90711998 90731997 90731998 90733998 90734998 91204998
91453997 91453998 91537997 91537998 91841998 92015998 92141998 92671998
92706998 92733998 92778998 92811990 92885997 92885998 92911990 93099997
93099998 93269992 93271992 93300998 93307990 93318998 93320998 93321998
93333998 93334997 93334998 93339998 93368998 93369997 93369998 93371998
93373997 93373998 93374996 93374997 93374998 93376998 93378998 93575998
93576998 93656997 93656998 93688992 93729992 93731992 93863998 93865992
93923992 94020992 94028990 94073992 94074992 94075992 94076992 94213998
94214997 94214998 94215996 94215997 94215998 94216997 94216998 94242998
94243998 94253998 94254997 94254998 94261998 94262998 94309990 94356997
94356998 94412992 94441990 94465990 94466990 94513997 94513998 94589997
94589998 94665992 94666992 94667992 94668992 94669992 94670992 94671992
94672992 94673992 94674992 94675992 94676992 94677992 94678992 94679992
94680992 94688990 94709996 94709997 94709998 94737992 94759998 95105990
95212992 95310990 95351990 95352990 95353990 95453998 95454998 95455998
95466998 95467997 95467998 95746992 95834997 95834998 95861992 95874998
95911992 96007990 96110992 96111992 96123990 96129990 96200990 96201990
96217996 96217997 96217998 96231992 96390990 96412998 96414990 96420990
96436990 96444990 96455990 96510998 96566989 96566990 96569992 96577992
96584992 96585992 96586992 96617997 96617998 96644992 96669998 96670997
96670998 96702998 96877992 96878992 96950992 96951990 96952992 96953992
96954992 96988989 96988990 96995989 96995990 96998990 97008990 97017989
97017990 97019989 97019990 97088992 97097990 97160990 97160998 97181990
97182990 97241989 97241990 97305992 97400998 97537989 97537990 97627992
97677989 97677990 97918989 97918990 97918998 97935989 97935990 97974989
97974990 98142989 98142990 98280998 98419996 98419997 98419998 98513989
98513990 98592988 98592989 98592990 98776996 98776997 98776998 98800989
98800990 98891989 98891990 99034998 99174998 99281990 99282988 99282989
99282990 99283989 99283990 99334996 99334997 99334998 99409998 99422998
99501998 99728998 99737992 99738992 99751992 99752992 99777992 99803992
99807988 99807989 99807990 99808988 99808989 99808990 99810988 99810989
99810990 99824998 99853998 99876992 99877992 99878992 99879992 99880992
99881992 99882992 99883992 99884992 99885992 99886992 99887992 99888992
99889992 99890992 99892992 99893992 99894992 99898992 99899992 99901992
99902992 99904992 99905992 {;
qui replace tx=17 if drugcode=="`x'";
};
***NSAID SCRIPTS***************************************************************;
#delimit ;
```

```
foreach x in 84433998 84435998 84490998 84553998 84554998 84555998 84556998
84586998 84758998 84833998 85154998 85428998 85741998 85783998 85784998
85848998 85849998 85953998 86209998 86533998 86534998 86594998 86624998
86628998 86629998 86635998 86886998 86953998 87070998 87144998 87145998
87243998 87244998 87245998 87246998 87247998 87248998 87413998 87495998
87593998 87595998 87978998 88012998 88021998 88022998 88042998 88047997
88047998 88092998 88093998 88094997 88094998 88127998 88130997 88130998
88133998 88138998 88139998 88145996 88145997 88174998 88228998 88233998
88284998 88294998 88295998 88296997 88296998 88309998 88414997 88414998
88442998 88455998 88456998 88464998 88520997 88520998 88527997 88527998
88746998 88817998 88840998 88881998 88882998 88894998 88904998 88943998
88970998 88977998 89010998 89014998 89022997 89022998 89028997 89028998
89052998 89117998 89137997 89137998 89158998 89162998 89167998 89171998
89176998 89217998 89302998 89336997 89336998 89344997 89344998 89368997
89368998 89390998 89395998 89398998 89404998 89405996 89405997 89405998
89419997 89419998 89420998 89462998 89464998 89465998 89466998 89479998
89484997 89484998 89511998 89516998 89533998 89558998 89572998 89578998
89580998 89593997 89593998 89621998 89691997 89691998 89745998 89760998
89784998 89785998 89801998 89852997 89852998 89890998 89909998 90072998
90080997 90080998 90116998 90119997 90119998 90125998 90151998 90303998
90342997 90342998 90346997 90346998 90351997 90351998 90358998 90360998
90361997 90361998 90368997 90368998 90512998 90628997 90628998 90635998
90636998 90793997 90793998 90846998 90869998 91049997 91049998 91081996
91081997 91081998 91082998 91091998 91105998 91109997 91109998 91120998
91144998 91203996 91203997 91203998 91213997 91213998 91269998 91414997
91414998 91420997 91420998 91421998 91434998 91463997 91463998 91502998
91519998 91581997 91581998 91583998 91616998 91621998 91747998 91774998
91815997 91815998 91877998 91920997 91920998 91988998 91989998 91990998
91991998 92000998 92112998 92113998 92121997 92121998 92122998 92158998
92169998 92184998 92185998 92189998 92194998 92210998 92224996 92224997
92224998 92290998 92368996 92368997 92368998 92384998 92519998 92604997
92604998 92650997 92650998 92652998 92668998 92706998 92801997 92801998
92850998 92851998 92862997 92862998 92863998 92864998 92950996 92950997
92950998 92951998 92952996 92952997 92952998 92953997 92953998 92954997
92954998 92964996 92964997 92964998 92965998 93005992 93006992 93029990
93029998 93048992 93072997 93072998 93085997 93085998 93086998 93089997
93089998 93110997 93110998 93135996 93135997 93135998 93152998 93169996
93169997 93169998 93170997 93170998 93195990 93196990 93197990 93215997
93215998 93216996 93216997 93216998 93217998 93218998 93247998 93261998
93262998 93267998 93272996 93272997 93272998 93351990 93352990 93353990
93368990 93369990 93390990 93441990 93442990 93459990 93481998 93488990
93489990 93533990 93534990 93538998 93540998 93549998 93570990 93571990
93579990 93580990 93590990 93605990 93606990 93669998 93697990 93698990
93710990 93711990 93714992 93721992 93725990 93726990 93756990 93793992
93866998 93951992 93988990 94016990 94022990 94028990 94029992 94030992
94031992 94086992 94163992 94165992 94166992 94198992 94214997 94240992
94254997 94254998 94258990 94259990 94259992 94287992 94309990 94342990
94343990 94347992 94352998 94362992 94370992 94410992 94427990 94428990
94441990 94458990 94459990 94460990 94489996 94489997 94489998 94491990
94492990 94514996 94514997 94514998 94515996 94515997 94515998 94524992
94539990 94540990 94540992 94614992 94614997 94614998 94623998 94626990
94627990 94651992 94654990 94655990 94659997 94659998 94670990 94678990
94679990 94707992 94743992 94750992 94755990 94784998 94790998 94798998
94805997 94805998 94809997 94809998 94832996 94832997 94832998 94838990
94874990 94874998 94875998 94886990 94887990 94887998 94892992 94901990
94907997 94907998 94914997 94914998 94928998 95014992 95017992 95024992
95061998 95075998 95093996 95093997 95093998 95143992 95146997 95146998
95157992 95167996 95167997 95167998 95168990 95169990 95176990 95227992
95227997 95227998 95233990 95235990 95247992 95259990 95260990 95266990
95289990 95300992 95303998 95312998 95313998 95340990 95347990 95348990
```

95359990 95360990 95366992 95367990 95391992 95397990 95496997 95496998
95497998 95498996 95498997 95498998 95538997 95538998 95539997 95539998
95540997 95540998 95541997 95541998 95542992 95546992 95567992 95633992
95634990 95634992 95645992 95646992 95647992 95673992 95754997 95754998
95833997 95833998 95852992 95909996 95909997 95909998 95921990 95981996
95981997 95981998 95993990 96035996 96035997 96035998 96085990 96098990
96108990 96109990 96126990 96126998 96127997 96127998 96128996 96128997
96128998 96135990 96136990 96138998 96139998 96140990 96140996 96140997
96140998 96141990 96142990 96238997 96238998 96278992 96298990 96310989
96310990 96341990 96362989 96362990 96366989 96366990 96369989 96369990
96369998 96385997 96385998 96399996 96399997 96399998 96400996 96400997
96400998 96404990 96405996 96405997 96405998 96407989 96407990 96418988
96418989 96418990 96426989 96426990 96442990 96451997 96451998 96452997
96452998 96466988 96466989 96466990 96495996 96495997 96495998 96531988
96531989 96531990 96540990 96550992 96557998 96558988 96558989 96558990
96558996 96558997 96558998 96583988 96583989 96625988 96625989 96625990
96641990 96728990 96762998 96763996 96763997 96763998 96789990 96802990
96815992 96834989 96834990 96841990 96842988 96842989 96842990 96845990
96851990 96867989 96867990 96879990 96881990 96890989 96890990 96921998
96930990 96936997 96936998 96939988 96939989 96939990 96955998 96959990
96961989 96961990 96966989 96966990 96984988 96984989 96984990 96988990
96994990 96996990 96997990 97000990 97002990 97004998 97005990 97022989
97022990 97024998 97028990 97049989 97049990 97056996 97056997 97056998
97083998 97100988 97100989 97100990 97102989 97102990 97104988 97104989
97104990 97105996 97105997 97105998 97106997 97106998 97107989 97107990
97107997 97107998 97111990 97112989 97112990 97114990 97126996 97126997
97126998 97138990 97147990 97156989 97156990 97180989 97180990 97181990
97187988 97187989 97187990 97188988 97188989 97188990 97203990 97209998
97210997 97210998 97211998 97212997 97212998 97228996 97228997 97228998
97229998 97230996 97230997 97230998 97231990 97240990 97295992 97355997
97355998 97356998 97357996 97357997 97357998 97358997 97358998 97420992
97447992 97483996 97483997 97483998 97537989 97543998 97550988 97550989
97550990 97551989 97551990 97565998 97566996 97566997 97566998 97593996
97593997 97593998 97594997 97594998 97657997 97657998 97658997 97658998
97668998 97674989 97674990 97678997 97678998 97682998 97700990 97704997
97704998 97712997 97712998 97714989 97714990 97734989 97734990 97746990
97748998 97753989 97753990 97756989 97756990 97765988 97765989 97765990
97769998 97800997 97800998 97801997 97801998 97809990 97810989 97810990
97811988 97811989 97811990 97817998 97825997 97825998 97833997 97833998
97902990 97906996 97906997 97906998 97923990 97932989 97932990 97935990
97959988 97959989 97959990 97963990 97965989 97965990 97985989 97985990
98018989 98018990 98040988 98040989 98040990 98041990 98060998 98077998
98078998 98084998 98096998 98098992 98127989 98127990 98134989 98134990
98150990 98151990 98166996 98166997 98166998 98322998 98335989 98335990
98346989 98346990 98347989 98347990 98357989 98357990 98364989 98364990
98399998 98400998 98426989 98426990 98429998 98495989 98495990 98504990
98515998 98516998 98526990 98528990 98529988 98529989 98529990 98530988
98530989 98530990 98555989 98555990 98569988 98569989 98569990 98578998
98600989 98600990 98604988 98604989 98604990 98621989 98621990 98628989
98628990 98629988 98629989 98629990 98654998 98671988 98671989 98671990
98672989 98672990 98673988 98673989 98673990 98674988 98674989 98674990
98675998 98689998 98690998 98691996 98691997 98691998 98692996 98692997
98692998 98693998 98758998 98763998 98764998 98779998 98796990 98907998
98932997 98932998 98970998 99040997 99040998 99093996 99093997 99093998
99153990 99184998 99213989 99213990 99232988 99232989 99232990 99233989
99233990 99233997 99233998 99234989 99234990 99234998 99292998 99301996
99301997 99301998 99302996 99302997 99302998 99303997 99303998 99334997
99334998 99410996 99410997 99410998 99442988 99442989 99442990 99443989
99443990 99444989 99444990 99445989 99445990 99466996 99466997 99466998
99482998 99487996 99487997 99487998 99516998 99517989 99517990 99518989

```
99518990 99519989 99519990 99520989 99520990 99535996 99535997 99535998
99539996 99539997 99539998 99550988 99550989 99550990 99551988 99551989
99551990 99552988 99552989 99552990 99553989 99553990 99557988 99557989
99557990 99558988 99558989 99558990 99621996 99621997 99621998 99644989
99644990 99644998 99653996 99653997 99653998 99727998 99728988 99728989
99728990 99729989 99729990 99730989 99730990 99730997 99730998 99731989
99731990 99743998 99760992 99788989 99788990 99789990 99823997 99823998
99868997 99868998 99903998 99970998 83880998 83984998 84128998 84129998
84151998 84152998 84153998 84155998 84160998 92812990 {;
qui replace tx=18 if drugcode=="`x'";
};
***SSRI SCRIPTS*****************************************************************;
#delimit ;
foreach x in 84436998 84807998 85382998 85970998 85971998 86159998 87249998
87250998 87251998 87662998 87663998 88285998 90159998 90766998 90814998
91380996 91380997 91380998 91395996 91395997 91395998 91671998 92172998
92174998 93042990 93066990 93173997 93173998 93174997 93174998 93374990
93375990 93475990 93476990 93487990 93489996 93489997 93489998 93490996
93490997 93490998 93693990 93694990 93701990 93702990 93728990 93729990
93732990 93733990 93735990 93736990 93737990 93738990 93739990 93740990
93741990 93742990 93744990 93745990 93746990 93747990 93748990 93749990
93752990 93753990 93790990 93813990 93818990 93834990 93837990 93842990
93843990 93905990 93946990 93947990 93948990 93994990 93995990 93996990
93997990 94212990 94231990 94232990 94242990 94260990 94261990 94262990
94420990 94447996 94447997 94447998 94490996 94490997 94490998 94602990
94603990 94604990 94760990 94836990 94852990 94853990 94873990 94880990
94881990 94882990 94893990 94894990 94895990 94925990 94936990 94937990
94938990 95007990 95028990 95051990 95141990 95269990 95270990 95271990
95332990 95333990 95334990 95335990 95350990 95388990 95395990 95418990
95420990 95421990 95426990 95529990 95578990 95607990 95610990 95631990
95632990 95633990 95666990 95667990 95668990 95703990 95704990 95705990
95813990 95820990 96087990 96092990 96093990 96241989 96241990 96272990
96281990 96345989 96345990 96492997 96492998 96493997 96493998 96606990
96643990 96644990 96647990 96651990 96654990 96659990 96674990 96709990
96729990 96810989 96810990 98088998 98561998 99592998 92840990 92841990
92845990 92929990 84403998 87334998 87335998 87336998 87337998 96534992
98327992 {;
qui replace tx=19 if drugcode=="`x'";
};
***WARFARIN SCRIPTS*************************************************************;
#delimit ;
foreach x in 84565998 86425998 88944998 92245998 92313998 93227990 93532990
93575990 93576990 93577990 94106990 94107990 94108990 94877990 94878990
94879990 95232990 95234990 95237990 95243992 95512990 95513990 95514990
95556996 95556997 95556998 95617996 95617997 95617998 95630990 95741992
96161990 96162990 96163990 96308988 96308989 96308990 96318988 96318989
96318990 96447988 96447989 96447990 96749997 96749998 97089988 97089989
97089990 97711988 97711989 97711990 97941988 97941989 97941990 98014988
98014989 98014990 98031988 98031989 98031990 98289996 98289997 98289998
98293996 98293997 98293998 98906996 98906997 98906998 99034988 99034989
99034990 99035989 99035990 99138997 99138998 99331988 99331989 99331990
83971998 83972998 83973998 83974998 83976998 83977998 {;
qui replace tx=20 if drugcode=="`x'";
};
***OTHER ANTIPLATLET SCRIPTS****************************************************;
#delimit ;
foreach x in 91304998 91305998 86450998 91406998 92084998 89385998 89393998
85500998 88829998 89165998 92941997 92941998 93672998 93821990 94031990
94032990 94142992 94160997 94160998 94431990 95615990 96183997 96183998
```

```
96340992 96343998 96344996 96344997 96344998 96817990 97763989 97763990
99225989 99225990 99226989 99226990 99227990 99228990 99263997 99263998
99618989 99618990 95400992 95401992 89968997 89968998 99378997 99378998
86449998 89136998 97794998 88855998 88856998 89035998 90355998 {;
qui replace tx=21 if drugcode=="`x'";
};
***HCV RX*************************************************************
#delimit ;
foreach x in 84993998 89105998 89271997 89565998 92300997 87415998 88134996
88134997 88134998 88306998 89172996 89172997 89172998 89173996 89173997
89173998 89334998 89436997 89436998 89612998 89613998 90735998 90936996
90936997 90936998 91133996 91133997 91134996 91134997 91134998 91135996
91135997 91136996 91136997 91136998 91379998 91382998 91383996 91383997
91383998 91599996 91599997 91599998 94701998 96117996 96117997 96117998
96118996 96118997 96118998 96119996 96119998 96120996 96121996 96121997
96121998 97054998 97055996 97055997 97055998 97852998 99979992 {;
qui replace tx=22 if drugcode=="`x'";
};
label define tx
1 beta
2 ccb
3 diur
4 vaso
5 cent
6 adre
7 alpha
8 ace
9 arb
10 renin
11 gang
12 insulin
13 sulph
14 biguan
15 other_dm
16 statin
17 asa
18 nsaid
19 ssri
20 warfarin
21 oth_antiplat
22 hcv_meds
;
#delimit cr
label values tx tx
compress
tab tx
drop if tx==0
sort pracid patid bmi_date tx
by pracid patid bmi_date tx: gen litn=_n
keep if litn==1
drop drugcode
reshape wide litn, i(pracid patid) j(tx)
rename litn1 beta
rename litn2 ccb
rename litn3 diur
rename litn4 vaso
rename litn5 cent
rename litn6 adre
rename litn7 alpha
rename litn8 ace
```

```
rename litn9 arb
rename litn10 renin
rename litn12 insulin
rename litn13 sulph
rename litn14 biguan
rename litn15 other_dm
rename litn16 statin
rename litn17 asa
rename litn18 nsaid
rename litn19 ssri
rename litn20 warfarin
rename litn21 oth_antiplat
rename litn22 hcv_meds
foreach x in beta ccb diur vaso cent adre alpha ace arb insulin sulph biguan
other_dm statin asa nsaid ssri warfarin oth_antiplat hcv_meds{
label var `x'
qui replace `x'=0 if `x'==.
}
compress
move bmi_date beta
sort pracid patid bmi_date
save "/Users/kforde/Desktop/THIN/pat_tx_ever.dta", replace

clear
use "/Users/kforde/Desktop/THIN/PVI_p10.dta"
drop urbrural eth_pcw eth_pcm eth_pcas eth_pcb eth_pco prp_llti no2 pm10 so2
nox upddate
order pracid patid townsend
sort pracid patid townsend
bysort pracid patid: generate n=_n
bysort pracid patid: generate N=_N
keep if n==N
replace townsend="." if townsend=="0"
replace townsend="." if townsend=="X"
destring townsend, replace
drop n N
***duplicates report pracid patid
sort pracid patid
save "/Users/kforde/Desktop/THIN/townsend.dta", replace

clear
use "/Users/kforde/Desktop/THIN/PVI_p10.dta"
drop eth_pcw eth_pcm eth_pcas eth_pcb eth_pco prp_llti no2 pm10 so2 nox
townsend upddate
order pracid patid urbrural
sort pracid patid urbrural
bysort pracid patid: generate n=_n
bysort pracid patid: generate N=_N
keep if n==N
replace urbrural="." if urbrural=="0"
destring urbrural, replace
drop n N
***duplicates report pracid patid
sort pracid patid
save "/Users/kforde/Desktop/THIN/urbrural.dta", replace
***GENERATING A FULL DEMOGRAPHICS FILE INCORPORATING THE SOCIOECONOMIC
INFORMATION***

clear
use "/Users/kforde/Desktop/THIN/comorbid_set_up.dta"
```

```
sort pracid patid bmi_date
merge 1:1 pracid patid using "/Users/kforde/Desktop/THIN/pat_comorbid_ever.dta"
foreach x in htn dm chol renal overweight obese alcohol hcv cocaine hbv cad
fxcad {
label var `x'
qui replace `x'=0 if `x'==.
}
drop if _merge==2
drop _merge
sort pracid patid bmi_date
merge 1:1 pracid patid using "/Users/kforde/Desktop/THIN/pat_tx_ever.dta"
drop if _merge==2
drop _merge
foreach x in beta ccb diur vaso cent adre alpha ace arb renin insulin sulph
biguan other_dm statin asa nsaid ssri warfarin oth_antiplat hcv_meds{
label var `x'
qui replace `x'=0 if `x'==.
}
sort pracid patid bmi_date
merge 1:1 pracid patid using "/Users/kforde/Desktop/THIN/townsend.dta"
drop if _merge==2
drop _merge
sort pracid patid bmi_date
merge 1:1 pracid patid using "/Users/kforde/Desktop/THIN/urbrural.dta"
drop if _merge==2
drop _merge
sort pracid patid bmi_date
save "/Users/kforde/Desktop/THIN/approach_1.dta", replace


*** Step 1, set a global to avoid edits to file folder if run on a different
computer
cd "/Users/kforde/Desktop/THIN"
clear all
*** Generate date specific log file
log using "test_app10.log", replace
*** Set maximum variable capacity and seed. Note seed changed for each run of
do file.
set maxvar 32767, perm
set seed 0000000001

*** Note the number of observations has to be set for the exact number of rows
in each of the subsequently created datasets.
set obs 20000
save results_app1.dta, replace emptyok

forvalues i = 001/100 {
       di "Iteration `i', Start time: $S_TIME"
       ***FORMATTING DEMOGRAPHICS FILE***
       clear
       clear mata
       drop _all
       use "Patdemo_p10.dta"
       ***GENERATION OF CALCULATED BMI VARIABLE***
       sort bmi
       qui generate height2=(height)*(height)
       qui generate calc_bmi=[(weight)/(height2)]
       recast double calc_bmi
       sum bmi, detail
       sum calc_bmi, detail
```

```
	***GENERATE ROUNDED BMI (CALCULATED)VARIABLE FOR COMPARISON BMI PROVIDED
IN DATASET
	qui generate bmi_round=round(bmi, 0.1)
	qui generate calc_bmi_round=round(calc_bmi, 0.1)
	count if bmi_round==.
	***72,354 missing observations
	count if bmi_round==. & calc_bmi_round==.
	***52,527 missing observations
	drop if bmi_round==. & calc_bmi_round==.
	***52,527 observations dropped
	***DROPPING MISSING/ INACCURATE BMI VALUES
	drop if weight==. | height==.
	***3 now observations dropped as calculation of bmi not possible
	drop if weight==0| height==0
	***247 observations dropped as calculation of bmi would be inaccurate
	****230,960 observations remain in the dataset
	drop if calc_bmi<10
	***1,070 observations deleted as bmi <10 not feasible
	***as per the THIN manual, their internal check for BMI is 13-100
	drop if calc_bmi>65
	***674 observations deleted as bmi>65 not "likely" feasible
	drop if height<=1.2
	***8,118 observations deleted as height <1.2 in a non-pediatric patient
is not feasible
	***as per THIN manual, their internal height check is 0.8-2.1. Weight is
0.5 - 180.
	drop if age<18
	***5,913 observations deleted as study only includes adults
	drop if age>90
	***5,408 observations deleted as upper limit of age for the study is 90
	count if bmi_round==calc_bmi_round
	count if calc_bmi_round==.
	***101,704 matches of calculated bmi to the bmi provided in THIN
	***DETERMINING SIGNIFICANT DISCPREPANCIES IN BMI CALCULATION WHEN
COMPARED TO BMI PROVIDED IN THIN
	qui generate tag=.
	qui replace tag=0 if bmi_round==calc_bmi_round
	qui replace tag=1 if bmi_round<calc_bmi_round+1 &
bmi_round>calc_bmi_round-1 & bmi_round!=calc_bmi_round
	qui replace tag=1 if bmi_round==(calc_bmi_round) + 1
	qui replace tag=1 if bmi_round==(calc_bmi_round) - 1
	qui replace tag=2 if bmi_round!=calc_bmi_round & tag==.
	qui replace tag=3 if bmi_round==.
	qui generate diff=bmi_round-calc_bmi_round
	***bysort tag: summ diff, detail
	***101,918 instances in which calculated bmi is within one unit of the
bmi provided in THIN
	***5,606 instances in which the bmi provided in THIN was missing
	***549 instances in which the discrepancies are extreme (0 to 308964.3)
	drop if tag==3
	drop if tag==2
	***As values of bmi were missing and some differences were largely
discrepant, 6,155 observations were dropped as the nature of the error was
unclear
	***GENERATION OF A BMI DATE (WEIGHT DATE USED AS HEIGHT UNLIKELY TO
CHANGE OVER TIME)
	qui generate ran=runiform(0,1)
	sort ran
	generate n=_n
	keep if n<=20000
```

```
	drop ran n
	qui generate bmi_date=wghtdate
	qui format bmi_date %td
	***20000 patients in the final demographics dataset

	***GENERATING MISSING COMPLETELY AT RANDOM BMI VALUES
	qui generate random_mcar=runiform(0,1)
	count if random_mcar<0.250
	qui generate bmi_mcar=calc_bmi
	qui replace bmi_mcar=. if random_mcar<0.250
	count if bmi_mcar==.
	qui compress
	***4,991 observations made missing = 0.24955 or 25% of the cohort - NOTE
THIS QUANITITY CHANGES SLIGHTLY EACH TIME WITH THE GENERATION OF A NEW RANDOM
NUMBER.

	***GENERATING CATEGORIES OF MISSING AT RANDOM BASED ON AGE AND GENDER
CATEGORIES
	run gen_miss_bmi_pattern.do  // runs do file that creates missingness

	***PREPARING THE MASTER DATASET***
	***FORMATTING VARIABLES***
	run gen_master.do // runs do file that creates master file setup

	***GENERATION OF DATASET WITH VARIOUS MISSING BMI VARIABLES (REFLECTING
MECHANISMS OF MISSINGNESS) AND MEDICAL AND THERAPY CODES
	clear
	use "bmis.dta", clear
	sort pracid patid bmi_date
	merge 1:1 pracid patid bmi_date using "approach_4.dta"
	keep if _merge==3
	drop _merge
	sort pracid patid bmi_date
	qui compress
	save "approach_new_4.dta", replace

	clear mata
	local mnar_settings mcar mar1 mar2 mar3 mar4 mnar1 mnar2_th mnar3_th
mnar4_3w
	foreach setting of local mnar_settings {
	use "approach_new_4.dta", clear
	merge 1:1 pracid patid bmi_date using demographics_plus_ut.dta
	qui keep bmi_`setting' pracid patid age birth_year gender urbrural
townsend start_date end_date transfer transfer_date death death_date bmi_date
htn dm chol renal alcohol_rc hcv cocaine hbv cad fxcad asa nsaid ssri warfarin
oth_antiplat
	qui sort pracid patid bmi_date
	qui compress
	save "`setting'_final_dataset_`i'.dta", replace
	}
	***MULTIPLE IMPUTATION 1000-ITERATION LOOPS***

	clear
	clear mata
	use "bmis.dta", clear
	qui sort pracid patid bmi_date
	qui keep pracid patid bmi_date calc_bmi_round
	qui compress
	save "bmi_truth.dta", replace
```

```
	local mnar_settings mcar mar1 mar2 mar3 mar4 mnar1 mnar2_th mnar3_th
mnar4_3w
	foreach setting of local mnar_settings {
		use "`setting'_final_dataset_`i'.dta", clear
		generate fu_time= (end_date-start_date)/365.25
		generate time_to_bmi= (bmi_date - start_date)/365.25
		mi set wide
		mi register imputed bmi_`setting' townsend urbrural
		mi register regular age gender transfer death htn dm chol renal
alcohol_rc hcv cocaine hbv cad fxcad asa nsaid ssri warfarin oth_antiplat
		mi register passive birth_year start_date end_date transfer_date
death_date bmi_date
		mi impute mvn bmi_`setting' townsend urbrural = age gender
transfer death fu_time time_to_bmi htn dm chol renal alcohol_rc hcv cocaine hbv
cad fxcad asa nsaid ssri warfarin oth_antiplat, add(10)
		sort pracid patid bmi_date
		merge 1:1 pracid patid bmi_date using "bmi_truth.dta"
		qui compress
		save "`setting'_final_dataset_`i'.dta", replace
		qui generate iteration=`i'
		qui generate bias1=[(_1_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias2=[(_2_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias3=[(_3_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias4=[(_4_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias5=[(_5_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias6=[(_6_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias7=[(_7_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias8=[(_8_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias9=[(_9_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias10=[(_10_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate
`setting'_bias_tot_`i'=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)
+(bias8)+(bias9)+(bias10)]/10
		***qui replace  `setting'_bias_tot_`i'=0 if _mi_miss==0
		qui keep `setting'_bias_tot_`i'
		qui merge 1:1 _n using results_app10, nogen
		save results_app10, replace
		}
	di "Iteration `i', End time: $S_TIME"
}
log close
```

```
/***PhD PROJECT 3: ASSESSMENT OF FEATURE SELECTION STRATEGIES FOR MULTIPLE
IMPUTATION/ VARIABLE SELECTION STRATEGY 2***
*/

*** Step 1, set a global
cd "/Users/kforde/Desktop/THIN"
clear all
*** Generate date specific log file
log using "XXX.log", replace
*** Set maximum variable capacity and seed. Seed changed for each run
set maxvar 32767, perm
set seed 0000000001

*** Note the number of observations has to be set for the exact number of rows
in each of the subsequently created datasets.
set obs 20000
save results.dta, replace emptyok

**Step 2, Set up of BMI – Gold standard
forvalues i = 001/100 {
	di "Iteration `i', Start time: $S_TIME"
	***FORMATTING DEMOGRAPHICS FILE***
	clear
	clear mata
	drop _all
	use "Patdemo_p10.dta"
	***GENERATION OF CALCULATED BMI VARIABLE***
	sort bmi
	qui generate height2=(height)*(height)
	qui generate calc_bmi=[(weight)/(height2)]
	recast double calc_bmi
	sum bmi, detail
	sum calc_bmi, detail
	***GENERATE CALCULATED BMI FOR COMPARISON TO BMI PROVIDED IN DATASET
	qui generate bmi_round=round(bmi, 0.1)
	qui generate calc_bmi_round=round(calc_bmi, 0.1)
	count if bmi_round==.
	***72,354 missing observations
	count if bmi_round==. & calc_bmi_round==.
	***52,527 missing observations
	drop if bmi_round==. & calc_bmi_round==.
	***52,527 observations dropped
	***DROPPING MISSING/ INACCURATE BMI VALUES
	drop if weight==. | height==.
	***3 now observations dropped as calculation of bmi not possible
	drop if weight==0| height==0
	***247 observations dropped as calculation of bmi would be inaccurate
	****230,960 observations remain in the dataset
	drop if calc_bmi<10
	***1,070 observations deleted as bmi <10 not feasible
	***as per the THIN manual, their internal check for BMI is 13-100
	drop if calc_bmi>65
	***674 observations deleted as bmi>65 not "likely" feasible
	drop if height<=1.2
	***8,118 observations deleted, height <1.2 in adult patient not feasible
	***as per THIN manual, internal height check 0.8-2.1/ Weight 0.5 - 180
	drop if age<18
	***5,913 observations deleted as study only includes adults
	drop if age>90
```

```
	***5,408 observations deleted as upper limit of age for the study was 90
	count if bmi_round==calc_bmi_round
	count if calc_bmi_round==.
	***101,704 matches of calculated bmi to the bmi provided in THIN
	***DETERMINING DISCPREPANCIES IN BMI CALCULATION COMPARED TO BMI IN THIN
	qui generate tag=.
	qui replace tag=0 if bmi_round==calc_bmi_round
	qui replace tag=1 if bmi_round<calc_bmi_round+1 &
bmi_round>calc_bmi_round-1 & bmi_round!=calc_bmi_round
	qui replace tag=1 if bmi_round==(calc_bmi_round) + 1
	qui replace tag=1 if bmi_round==(calc_bmi_round) - 1
	qui replace tag=2 if bmi_round!=calc_bmi_round & tag==.
	qui replace tag=3 if bmi_round==.
	qui generate diff=bmi_round-calc_bmi_round
	***101,918 calculated bmi is within one unit of the bmi provided in THIN
	***5,606 instances in which the bmi provided in THIN was missing
	***549 instances in which the discrepancies are extreme (0 to 308964.3)
	drop if tag==3
	drop if tag==2
	***As values of bmi were missing and some differences were largely
discrepant, 6,155 observations were dropped as the nature of the error was
unclear
	***GENERATION OF A BMI DATE(WEIGHT DATE USED, NO HEIGHT CHANGE OVER TIME)
	qui generate ran=runiform(0,1)
	sort ran
	generate n=_n
	keep if n<=20000
	drop ran n
	qui generate bmi_date=wghtdate
	qui format bmi_date %td
	***20000 patients in the final analytic dataset
**Step 3: Generate mechanisms of missingness
***GENERATING MECHANISMS OF MISSING
	run gen_miss_bmi_pattern.do

**Step 4: Master dataset generation
	***PREPARING THE MASTER DATASET***
	run gen_master.do

	***GENERATION OF DATASET WITH VARIOUS MISSING BMI VARIABLES (REFLECTING
MECHANISMS OF MISSINGNESS) AND MEDICAL AND THERAPY CODES
	clear
	use "bmis.dta", clear
	sort pracid patid bmi_date
	merge 1:1 pracid patid bmi_date using "code_bmi_dataset.dta"
	keep if _merge==3
	drop _merge
	sort pracid patid bmi_date
	qui compress
	save "code_bmi_dataset_new.dta", replace

**Step 5, Evaluating statistical threshold for inclusion of variables
	***GENERATION OF CORRELATION COEFFICIENTS FOR THE BMI VARIABLES AND
GENERATION OF FINAL DATASETS FOR MULTIPLE IMPUTATION******
	***EVEN THOUGH THIS CODE IS MODIFIED TO RUN QUIETLY, IT STILL TAKES HOURS
FOR EACH SECTION TO RUN

	clear mata
	local mnar_settings mcar mar1 mar2 mar3 mar4 mnar1 mnar2_th mnar3_th
mnar4_3w
```

```
	foreach setting of local mnar_settings {
		clear
		  capture erase "kforde_`setting'_1.dta"
		   use "code_bmi_dataset_new.dta", clear
				foreach v of varlist code* {
					capture quietly pwcorr bmi_`setting' `v', sig
					qui gen r_`v'=abs(r(rho))
					qui if r_`v' <0.1 drop `v'
					qui if r_`v' ==. drop `v'
					qui drop r_`v'
			 }
			merge 1:1 pracid patid bmi_date using
demographics_plus_ut.dta
			qui keep bmi_`setting' pracid patid age birth_year gender
urbrural townsend start_date end_date transfer transfer_date death death_date
bmi_date code*
			qui sort pracid patid bmi_date
			qui compress
			save "`setting'_final_dataset_`i'.dta", replace
		}

**Step 6, Multiple imputation
	***MULTIPLE IMPUTATION 1000-ITERATION LOOPS***

	clear
	clear mata
	use "bmis.dta", clear
	qui sort pracid patid bmi_date
	qui keep pracid patid bmi_date calc_bmi_round
	qui compress
	save "bmi_truth.dta", replace

	local mnar_settings mcar mar1 mar2 mar3 mar4 mnar1 mnar2_th mnar3_th
mnar4_3w
	foreach setting of local mnar_settings {
		use `setting'_final_dataset_`i'.dta, clear
		generate fu_time= (end_date-start_date)/365.25
		generate time_to_bmi= (bmi_date - start_date)/365.25
		mi set wide
		mi register imputed bmi_`setting' urbrural townsend
		mi register regular age gender transfer death code*
		mi register passive birth_year start_date end_date transfer_date
death_date bmi_date
		mi impute mvn bmi_`setting' urbrural townsend = age i.gender
i.transfer i.death fu_time time_to_bmi code*, add(10)
		sort pracid patid bmi_date
		merge 1:1 pracid patid bmi_date using "bmi_truth.dta",  nogen

*Step 7, Generating the results for bias/ percent bias
		qui generate iteration=`i'
		qui generate bias1=[(_1_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias2=[(_2_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias3=[(_3_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias4=[(_4_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
		qui generate bias5=[(_5_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
```

```
             qui generate bias6=[(_6_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
             qui generate bias7=[(_7_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
             qui generate bias8=[(_8_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
             qui generate bias9=[(_9_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
             qui generate bias10=[(_10_bmi_`setting')-
(calc_bmi_round)]/calc_bmi_round if _mi_miss==1 & bmi_`setting'==.
             qui generate
`setting'_bias_tot_`i'=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)
+(bias8)+(bias9)+(bias10)]/10
             ***qui replace  `setting'_bias_tot_`i'=0 if _mi_miss==0
             qui keep `setting'_bias_tot_`i'
             qui merge 1:1 _n using results, nogen
             save results, replace
             }
      di "Iteration `i', End time: $S_TIME"
}
log close
```

Appendix Item 4.2: STATA Do-File to Create MCAR, MAR and MNAR Missingness Mechanism Patterns

```
*** MISSING COMPLETELY AT RANDOM SET UP
set seed 12345
gen random_mcar=runiform(0,1)
count if random_mcar<0.250
gen bmi_mcar=calc_bmi
replace bmi_mcar=. if random_mcar<0.250


***MISSING AT RANDOM SET UP (BASED ON AGE AND GENDER ASSOCIATION WITH MISSING)
sum age, detail
tab gender
generate random_cat=0
replace random_cat=1 if age<55 & gender==0
replace random_cat=2 if age<55 & gender==1
replace random_cat=3 if age>=55 & gender==0
replace random_cat=4 if age>=55 & gender==1
tab random_cat
generate random_cat_1=runiform(0,1) if random_cat==1
generate random_cat_2=runiform(0,1) if random_cat==2
generate random_cat_3=runiform(0,1) if random_cat==3
generate random_cat_4=runiform(0,1) if random_cat==4
generate random_mar=.
replace random_mar=random_cat_1 if random_cat_1!=.
replace random_mar=random_cat_2 if random_cat_2!=.
replace random_mar=random_cat_3 if random_cat_3!=.
replace random_mar=random_cat_4 if random_cat_4!=.
generate bmi_mar1=calc_bmi
replace bmi_mar1=. if random_cat_1!=. & random_cat_1<0.2
replace bmi_mar1=. if random_cat_2!=. & random_cat_2<0.1
replace bmi_mar1=. if random_cat_3!=. & random_cat_3<0.4
replace bmi_mar1=. if random_cat_4!=. & random_cat_4<0.3
count if bmi_mar1==.
***Missing data pattern 1: men>women and older>younger
generate bmi_mar2=calc_bmi
replace bmi_mar2=. if random_cat_1!=. & random_cat_1<0.4
replace bmi_mar2=. if random_cat_2!=. & random_cat_2<0.3
replace bmi_mar2=. if random_cat_3!=. & random_cat_3<0.2
replace bmi_mar2=. if random_cat_4!=. & random_cat_4<0.1
count if bmi_mar2==.
***Missing data pattern 2: men>women and younger>older
generate bmi_mar3=calc_bmi
replace bmi_mar3=. if random_cat_1!=. & random_cat_1<0.1
replace bmi_mar3=. if random_cat_2!=. & random_cat_2<0.2
replace bmi_mar3=. if random_cat_3!=. & random_cat_3<0.3
replace bmi_mar3=. if random_cat_4!=. & random_cat_4<0.4
count if bmi_mar3==.
***Missing data pattern 3: women>men and older>younger
generate bmi_mar4=calc_bmi
replace bmi_mar4=. if random_cat_1!=. & random_cat_1<0.2
replace bmi_mar4=. if random_cat_2!=. & random_cat_2<0.4
replace bmi_mar4=. if random_cat_3!=. & random_cat_3<0.1
replace bmi_mar4=. if random_cat_4!=. & random_cat_4<0.2
count if bmi_mar4==.
```

```
***Missing data pattern 4: women>men and younger>older

***GENERATING MISSING NOT AT RANDOM: BASED ON CATEGORIES OF BMI
generate bmi_cat=.
replace bmi_cat=1 if calc_bmi>=10 & calc_bmi<15
replace bmi_cat=2 if calc_bmi>=15 & calc_bmi<20
replace bmi_cat=3 if calc_bmi>=20 & calc_bmi<25
replace bmi_cat=4 if calc_bmi>=25 & calc_bmi<30
replace bmi_cat=5 if calc_bmi>=30 & calc_bmi<35
replace bmi_cat=6 if calc_bmi>=35 & calc_bmi<40
replace bmi_cat=7 if calc_bmi>=40 & calc_bmi<45
replace bmi_cat=8 if calc_bmi>=45 & calc_bmi<50
replace bmi_cat=9 if calc_bmi>=50 & calc_bmi<55
replace bmi_cat=10 if calc_bmi>=55 & calc_bmi<60
replace bmi_cat=11 if calc_bmi>=60 & calc_bmi<=65
tab bmi_cat
generate random_bmi_cat_1=runiform(0,1) if bmi_cat==1
generate random_bmi_cat_2=runiform(0,1) if bmi_cat==2
generate random_bmi_cat_3=runiform(0,1) if bmi_cat==3
generate random_bmi_cat_4=runiform(0,1) if bmi_cat==4
generate random_bmi_cat_5=runiform(0,1) if bmi_cat==5
generate random_bmi_cat_6=runiform(0,1) if bmi_cat==6
generate random_bmi_cat_7=runiform(0,1) if bmi_cat==7
generate random_bmi_cat_8=runiform(0,1) if bmi_cat==8
generate random_bmi_cat_9=runiform(0,1) if bmi_cat==9
generate random_bmi_cat_10=runiform(0,1) if bmi_cat==10
generate random_bmi_cat_11=runiform(0,1) if bmi_cat==11
generate random_mnar=.
replace random_mnar=random_bmi_cat_1 if random_bmi_cat_1!=.
replace random_mnar=random_bmi_cat_2 if random_bmi_cat_2!=.
replace random_mnar=random_bmi_cat_3 if random_bmi_cat_3!=.
replace random_mnar=random_bmi_cat_4 if random_bmi_cat_4!=.
replace random_mnar=random_bmi_cat_5 if random_bmi_cat_5!=.
replace random_mnar=random_bmi_cat_6 if random_bmi_cat_6!=.
replace random_mnar=random_bmi_cat_7 if random_bmi_cat_7!=.
replace random_mnar=random_bmi_cat_8 if random_bmi_cat_8!=.
replace random_mnar=random_bmi_cat_9 if random_bmi_cat_9!=.
replace random_mnar=random_bmi_cat_10 if random_bmi_cat_10!=.
replace random_mnar=random_bmi_cat_11 if random_bmi_cat_11!=.
generate bmi_mnar1=calc_bmi
replace bmi_mnar1=. if random_mnar<0.10 & random_bmi_cat_1!=.
replace bmi_mnar1=. if random_mnar<0.15 & random_bmi_cat_2!=.
replace bmi_mnar1=. if random_mnar<0.20 & random_bmi_cat_3!=.
replace bmi_mnar1=. if random_mnar<0.25 & random_bmi_cat_4!=.
replace bmi_mnar1=. if random_mnar<0.30 & random_bmi_cat_5!=.
replace bmi_mnar1=. if random_mnar<0.35 & random_bmi_cat_6!=.
replace bmi_mnar1=. if random_mnar<0.40 & random_bmi_cat_7!=.
replace bmi_mnar1=. if random_mnar<0.45 & random_bmi_cat_8!=.
replace bmi_mnar1=. if random_mnar<0.50 & random_bmi_cat_9!=.
replace bmi_mnar1=. if random_mnar<0.55 & random_bmi_cat_10!=.
replace bmi_mnar1=. if random_mnar<0.60 & random_bmi_cat_11!=.
count if bmi_mnar1==.
***Missing data pattern 1: Increasing missing for each increase in bmi category
generate bmi_mnar2_th=calc_bmi
replace bmi_mnar2_th=. if random_mnar<0.75 & random_bmi_cat_5!=.
replace bmi_mnar2_th=. if random_mnar<0.80 & random_bmi_cat_6!=.
replace bmi_mnar2_th=. if random_mnar<0.85 & random_bmi_cat_7!=.
replace bmi_mnar2_th=. if random_mnar<0.90 & random_bmi_cat_8!=.
replace bmi_mnar2_th=. if random_mnar<0.90 & random_bmi_cat_9!=.
replace bmi_mnar2_th=. if random_mnar<0.90 & random_bmi_cat_10!=.
```

```
replace bmi_mnar2_th=. if random_mnar<0.90 & random_bmi_cat_11!=.
count if bmi_mnar2_th==.
***Missing data pattern 2: Threshold of missingness beginning at BMI of 30
generate bmi_mnar3_th=calc_bmi
replace bmi_mnar3_th=. if random_mnar<0.65 & random_bmi_cat_1!=.
replace bmi_mnar3_th=. if random_mnar<0.55 & random_bmi_cat_2!=.
replace bmi_mnar3_th=. if random_mnar<0.35 & random_bmi_cat_3!=.
replace bmi_mnar3_th=. if random_mnar<0.15 & random_bmi_cat_4!=.
count if bmi_mnar3_th==.
***Missing data pattern 3: Threshold of missingness from lowest BMI to 30
generate bmi_mnar4_3w=calc_bmi
replace bmi_mnar4_3w=. if random_mnar<0.65 & random_bmi_cat_1!=.
replace bmi_mnar4_3w=. if random_mnar<0.55 & random_bmi_cat_2!=.
replace bmi_mnar4_3w=. if random_mnar<0.20 & random_bmi_cat_3!=.
replace bmi_mnar4_3w=. if random_mnar<0.20 & random_bmi_cat_4!=.
replace bmi_mnar4_3w=. if random_mnar<0.20 & random_bmi_cat_5!=.
replace bmi_mnar4_3w=. if random_mnar<0.20 & random_bmi_cat_6!=.
replace bmi_mnar4_3w=. if random_mnar<0.50 & random_bmi_cat_7!=.
replace bmi_mnar4_3w=. if random_mnar<0.55 & random_bmi_cat_8!=.
replace bmi_mnar4_3w=. if random_mnar<0.60 & random_bmi_cat_9!=.
replace bmi_mnar4_3w=. if random_mnar<0.65 & random_bmi_cat_10!=.
replace bmi_mnar4_3w=. if random_mnar<0.70 & random_bmi_cat_11!=.
count if bmi_mnar4_3w==.
***Missing data pattern 4: Missingness most pronounced at ends of distribution
```

Appendix Item 4.3: STATA Do-File to Merge Final Datasets/Final Data Files

```
clear
cd "/Volumes/Thesis/High Dimensional/New MI Datasets"
log using "results_final_11_28_final.log", replace
  use mcar_final_dataset_1_mi.dta, clear
       foreach num of numlist 2/1000{
          qui append using mcar_final_dataset_`num'_mi.dta
               }

save mcar_final_large.dta, replace
keep pracid patid bmi_mcar calc_bmi_round _mi_miss _1_bmi_mcar _2_bmi_mcar
_3_bmi_mcar _4_bmi_mcar _5_bmi_mcar _6_bmi_mcar _7_bmi_mcar _8_bmi_mcar
_9_bmi_mcar _10_bmi_mcar
save mcar_final_short.dta, replace

qui generate bias1=[(_1_bmi_mcar)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mcar==.
qui generate bias2=[(_2_bmi_mcar)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mcar==.
qui generate bias3=[(_3_bmi_mcar)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mcar==.
qui generate bias4=[(_4_bmi_mcar)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mcar==.
qui generate bias5=[(_5_bmi_mcar)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mcar==.
qui generate bias6=[(_6_bmi_mcar)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mcar==.
qui generate bias7=[(_7_bmi_mcar)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mcar==.
qui generate bias8=[(_8_bmi_mcar)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mcar==.
qui generate bias9=[(_9_bmi_mcar)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mcar==.
qui generate bias10=[(_10_bmi_mcar)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mcar==.
qui generate
mcar_bias_tot_1=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)+(bias8
)+(bias9)+(bias10)]/10
summ mcar_bias_tot_1, detail
ci means mcar_bias_tot_1

qui generate error_1=[(_1_bmi_mcar)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mcar==.
qui generate error_2=[(_2_bmi_mcar)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mcar==.
qui generate error_3=[(_3_bmi_mcar)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mcar==.
qui generate error_4=[(_4_bmi_mcar)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mcar==.
qui generate error_5=[(_5_bmi_mcar)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mcar==.
qui generate error_6=[(_6_bmi_mcar)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mcar==.
qui generate error_7=[(_7_bmi_mcar)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mcar==.
qui generate error_8=[(_8_bmi_mcar)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mcar==.
```

```
qui generate error_9=[(_9_bmi_mcar)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mcar==.
qui generate error_10=[(_10_bmi_mcar)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mcar==.

qui generate n_mse=[(error_1) + (error_2) + (error_3) + (error_4) + (error_5) +
(error_6) + (error_7) + (error_8) + (error_9) + (error_10)]/10
summ n_mse, detail
ci means n_mse

qui generate error1=[(_1_bmi_mcar)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mcar==.
qui generate error2=[(_2_bmi_mcar)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mcar==.
qui generate error3=[(_3_bmi_mcar)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mcar==.
qui generate error4=[(_4_bmi_mcar)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mcar==.
qui generate error5=[(_5_bmi_mcar)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mcar==.
qui generate error6=[(_6_bmi_mcar)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mcar==.
qui generate error7=[(_7_bmi_mcar)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mcar==.
qui generate error8=[(_8_bmi_mcar)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mcar==.
qui generate error9=[(_9_bmi_mcar)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mcar==.
qui generate error10=[(_10_bmi_mcar)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mcar==.

qui generate m_error=[(error1) + (error2) + (error3) + (error4) + (error5) +
(error6) + (error7) + (error8) + (error9) + (error10)]/10
summ m_error, detail
ci means m_error

qui generate mse=[m_error*m_error]
summ mse, detail
ci means mse

qui generate cc1=.
qui replace cc1=1 if _1_bmi_mcar<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc1=1 if _1_bmi_mcar>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc1=0 if _mi_miss==1 & cc1==. & bmi_mcar==.
qui generate cc2=.
qui replace cc2=1 if _2_bmi_mcar<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc2=1 if _2_bmi_mcar>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc2=0 if _mi_miss==1 & cc2==. & bmi_mcar==.
qui generate cc3=.
qui replace cc3=1 if _3_bmi_mcar<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc3=1 if _3_bmi_mcar>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc3=0 if _mi_miss==1 & cc3==. & bmi_mcar==.
qui generate cc4=.
```

```
qui replace cc4=1 if _4_bmi_mcar<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc4=1 if _4_bmi_mcar>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc4=0 if _mi_miss==1 & cc4==. & bmi_mcar==.
qui generate cc5=.
qui replace cc5=1 if _5_bmi_mcar<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc5=1 if _5_bmi_mcar>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc5=0 if _mi_miss==1 & cc5==. & bmi_mcar==.
qui generate cc6=.
qui replace cc6=1 if _6_bmi_mcar<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc6=1 if _6_bmi_mcar>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc6=0 if _mi_miss==1 & cc6==. & bmi_mcar==.
qui generate cc7=.
qui replace cc7=1 if _7_bmi_mcar<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc7=1 if _7_bmi_mcar>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc7=0 if _mi_miss==1 & cc7==. & bmi_mcar==.
qui generate cc8=.
qui replace cc8=1 if _8_bmi_mcar<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc8=1 if _8_bmi_mcar>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc8=0 if _mi_miss==1 & cc8==. & bmi_mcar==.
qui generate cc9=.
qui replace cc9=1 if _9_bmi_mcar<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc9=1 if _9_bmi_mcar>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc9=0 if _mi_miss==1 & cc9==. & bmi_mcar==.
qui generate cc10=.
qui replace cc10=1 if _10_bmi_mcar<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc10=1 if _10_bmi_mcar>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mcar==.
qui replace cc10=0 if _mi_miss==1 & cc10==. & bmi_mcar==.
qui generate percent_cc=[cc1 + cc2 + cc3 + cc4 + cc5 + cc6 + cc7 + cc8 + cc9 +
cc10]/10
summ percent_cc, detail
ci means percent_cc


clear
cd "/Volumes/Thesis/High Dimensional/New MI Datasets"
  use mar1_final_dataset_1_mi.dta, clear
       foreach num of numlist 2/1000 {
          qui append using mar1_final_dataset_`num'_mi.dta
               }

save mar1_final_large.dta, replace
keep pracid patid bmi_mar1 calc_bmi_round _mi_miss _1_bmi_mar1 _2_bmi_mar1
_3_bmi_mar1 _4_bmi_mar1 _5_bmi_mar1 _6_bmi_mar1 _7_bmi_mar1 _8_bmi_mar1
_9_bmi_mar1 _10_bmi_mar1
save mar1_final_short.dta, replace
```

```
qui generate bias1=[(_1_bmi_mar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar1==.
qui generate bias2=[(_2_bmi_mar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar1==.
qui generate bias3=[(_3_bmi_mar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar1==.
qui generate bias4=[(_4_bmi_mar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar1==.
qui generate bias5=[(_5_bmi_mar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar1==.
qui generate bias6=[(_6_bmi_mar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar1==.
qui generate bias7=[(_7_bmi_mar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar1==.
qui generate bias8=[(_8_bmi_mar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar1==.
qui generate bias9=[(_9_bmi_mar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar1==.
qui generate bias10=[(_10_bmi_mar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar1==.
qui generate
mar1_bias_tot_1=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)+(bias8
)+(bias9)+(bias10)]/10
summ mar1_bias_tot_1, detail
ci means mar1_bias_tot_1

qui generate error_1=[(_1_bmi_mar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar1==.
qui generate error_2=[(_2_bmi_mar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar1==.
qui generate error_3=[(_3_bmi_mar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar1==.
qui generate error_4=[(_4_bmi_mar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar1==.
qui generate error_5=[(_5_bmi_mar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar1==.
qui generate error_6=[(_6_bmi_mar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar1==.
qui generate error_7=[(_7_bmi_mar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar1==.
qui generate error_8=[(_8_bmi_mar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar1==.
qui generate error_9=[(_9_bmi_mar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar1==.
qui generate error_10=[(_10_bmi_mar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar1==.

qui generate n_mse=[(error_1) + (error_2) + (error_3) + (error_4) + (error_5) +
(error_6) + (error_7) + (error_8) + (error_9) + (error_10)]/10
summ n_mse, detail
ci means n_mse

qui generate error1=[(_1_bmi_mar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar1==.
qui generate error2=[(_2_bmi_mar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar1==.
qui generate error3=[(_3_bmi_mar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar1==.
qui generate error4=[(_4_bmi_mar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar1==.
```

```
qui generate error5=[(_5_bmi_mar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar1==.
qui generate error6=[(_6_bmi_mar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar1==.
qui generate error7=[(_7_bmi_mar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar1==.
qui generate error8=[(_8_bmi_mar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar1==.
qui generate error9=[(_9_bmi_mar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar1==.
qui generate error10=[(_10_bmi_mar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar1==.

qui generate m_error=[(error1) + (error2) + (error3) + (error4) + (error5) +
(error6) + (error7) + (error8) + (error9) + (error10)]/10
summ m_error, detail
ci means m_error

qui generate mse=[m_error*m_error]
summ mse, detail
ci means mse

qui generate cc1=.
qui replace cc1=1 if _1_bmi_mar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc1=1 if _1_bmi_mar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc1=0 if _mi_miss==1 & cc1==. & bmi_mar1==.
qui generate cc2=.
qui replace cc2=1 if _2_bmi_mar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc2=1 if _2_bmi_mar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc2=0 if _mi_miss==1 & cc2==. & bmi_mar1==.
qui generate cc3=.
qui replace cc3=1 if _3_bmi_mar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc3=1 if _3_bmi_mar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc3=0 if _mi_miss==1 & cc3==. & bmi_mar1==.
qui generate cc4=.
qui replace cc4=1 if _4_bmi_mar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc4=1 if _4_bmi_mar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc4=0 if _mi_miss==1 & cc4==. & bmi_mar1==.
qui generate cc5=.
qui replace cc5=1 if _5_bmi_mar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc5=1 if _5_bmi_mar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc5=0 if _mi_miss==1 & cc5==. & bmi_mar1==.
qui generate cc6=.
qui replace cc6=1 if _6_bmi_mar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc6=1 if _6_bmi_mar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc6=0 if _mi_miss==1 & cc6==. & bmi_mar1==.
qui generate cc7=.
```

```
qui replace cc7=1 if _7_bmi_mar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc7=1 if _7_bmi_mar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc7=0 if _mi_miss==1 & cc7==. & bmi_mar1==.
qui generate cc8=.
qui replace cc8=1 if _8_bmi_mar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc8=1 if _8_bmi_mar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc8=0 if _mi_miss==1 & cc8==. & bmi_mar1==.
qui generate cc9=.
qui replace cc9=1 if _9_bmi_mar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc9=1 if _9_bmi_mar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc9=0 if _mi_miss==1 & cc9==. & bmi_mar1==.
qui generate cc10=.
qui replace cc10=1 if _10_bmi_mar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc10=1 if _10_bmi_mar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar1==.
qui replace cc10=0 if _mi_miss==1 & cc10==. & bmi_mar1==.
qui generate percent_cc=[cc1 + cc2 + cc3 + cc4 + cc5 + cc6 + cc7 + cc8 + cc9 +
cc10]/10
summ percent_cc, detail
ci means percent_cc

clear
cd "/Volumes/Thesis/High Dimensional/New MI Datasets"
  use mar2_final_dataset_1_mi.dta, clear
       foreach num of numlist 2/1000 {
          qui append using mar2_final_dataset_`num'_mi.dta
               }

save mar2_final_large.dta, replace
keep pracid patid bmi_mar2 calc_bmi_round _mi_miss _1_bmi_mar2 _2_bmi_mar2
_3_bmi_mar2 _4_bmi_mar2 _5_bmi_mar2 _6_bmi_mar2 _7_bmi_mar2 _8_bmi_mar2
_9_bmi_mar2 _10_bmi_mar2
save mar2_final_short.dta, replace

qui generate bias1=[(_1_bmi_mar2)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar2==.
qui generate bias2=[(_2_bmi_mar2)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar2==.
qui generate bias3=[(_3_bmi_mar2)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar2==.
qui generate bias4=[(_4_bmi_mar2)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar2==.
qui generate bias5=[(_5_bmi_mar2)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar2==.
qui generate bias6=[(_6_bmi_mar2)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar2==.
qui generate bias7=[(_7_bmi_mar2)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar2==.
qui generate bias8=[(_8_bmi_mar2)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar2==.
qui generate bias9=[(_9_bmi_mar2)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar2==.
```

```
qui generate bias10=[(_10_bmi_mar2)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar2==.
qui generate
mar2_bias_tot_1=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)+(bias8
)+(bias9)+(bias10)]/10
summ mar2_bias_tot_1, detail
ci means mar2_bias_tot_1

qui generate error_1=[(_1_bmi_mar2)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar2==.
qui generate error_2=[(_2_bmi_mar2)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar2==.
qui generate error_3=[(_3_bmi_mar2)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar2==.
qui generate error_4=[(_4_bmi_mar2)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar2==.
qui generate error_5=[(_5_bmi_mar2)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar2==.
qui generate error_6=[(_6_bmi_mar2)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar2==.
qui generate error_7=[(_7_bmi_mar2)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar2==.
qui generate error_8=[(_8_bmi_mar2)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar2==.
qui generate error_9=[(_9_bmi_mar2)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar2==.
qui generate error_10=[(_10_bmi_mar2)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar2==.

qui generate n_mse=[(error_1) + (error_2) + (error_3) + (error_4) + (error_5) +
(error_6) + (error_7) + (error_8) + (error_9) + (error_10)]/10
summ n_mse, detail
ci means n_mse

qui generate error1=[(_1_bmi_mar2)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar2==.
qui generate error2=[(_2_bmi_mar2)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar2==.
qui generate error3=[(_3_bmi_mar2)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar2==.
qui generate error4=[(_4_bmi_mar2)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar2==.
qui generate error5=[(_5_bmi_mar2)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar2==.
qui generate error6=[(_6_bmi_mar2)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar2==.
qui generate error7=[(_7_bmi_mar2)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar2==.
qui generate error8=[(_8_bmi_mar2)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar2==.
qui generate error9=[(_9_bmi_mar2)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar2==.
qui generate error10=[(_10_bmi_mar2)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar2==.

qui generate m_error=[(error1) + (error2) + (error3) + (error4) + (error5) +
(error6) + (error7) + (error8) + (error9) + (error10)]/10
summ m_error, detail
ci means m_error
```

```
qui generate mse=[m_error*m_error]
summ mse, detail
ci means mse

qui generate cc1=.
qui replace cc1=1 if _1_bmi_mar2<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc1=1 if _1_bmi_mar2>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc1=0 if _mi_miss==1 & cc1==. & bmi_mar2==.
qui generate cc2=.
qui replace cc2=1 if _2_bmi_mar2<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc2=1 if _2_bmi_mar2>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc2=0 if _mi_miss==1 & cc2==. & bmi_mar2==.
qui generate cc3=.
qui replace cc3=1 if _3_bmi_mar2<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc3=1 if _3_bmi_mar2>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc3=0 if _mi_miss==1 & cc3==. & bmi_mar2==.
qui generate cc4=.
qui replace cc4=1 if _4_bmi_mar2<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc4=1 if _4_bmi_mar2>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc4=0 if _mi_miss==1 & cc4==. & bmi_mar2==.
qui generate cc5=.
qui replace cc5=1 if _5_bmi_mar2<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc5=1 if _5_bmi_mar2>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc5=0 if _mi_miss==1 & cc5==. & bmi_mar2==.
qui generate cc6=.
qui replace cc6=1 if _6_bmi_mar2<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc6=1 if _6_bmi_mar2>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc6=0 if _mi_miss==1 & cc6==. & bmi_mar2==.
qui generate cc7=.
qui replace cc7=1 if _7_bmi_mar2<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc7=1 if _7_bmi_mar2>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc7=0 if _mi_miss==1 & cc7==. & bmi_mar2==.
qui generate cc8=.
qui replace cc8=1 if _8_bmi_mar2<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc8=1 if _8_bmi_mar2>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc8=0 if _mi_miss==1 & cc8==. & bmi_mar2==.
qui generate cc9=.
qui replace cc9=1 if _9_bmi_mar2<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc9=1 if _9_bmi_mar2>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc9=0 if _mi_miss==1 & cc9==. & bmi_mar2==.
qui generate cc10=.
```

```
qui replace cc10=1 if _10_bmi_mar2<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc10=1 if _10_bmi_mar2>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar2==.
qui replace cc10=0 if _mi_miss==1 & cc10==. & bmi_mar2==.
qui generate percent_cc=[cc1 + cc2 + cc3 + cc4 + cc5 + cc6 + cc7 + cc8 + cc9 +
cc10]/10
summ percent_cc, detail
ci means percent_cc

clear
cd "/Volumes/Thesis/High Dimensional/New MI Datasets"
  use mar3_final_dataset_1_mi.dta, clear
       foreach num of numlist 2/1000 {
          qui append using mar3_final_dataset_`num'_mi.dta
               }
save mar3_final_large.dta, replace
keep pracid patid bmi_mar3 calc_bmi_round _mi_miss _1_bmi_mar3 _2_bmi_mar3
_3_bmi_mar3 _4_bmi_mar3 _5_bmi_mar3 _6_bmi_mar3 _7_bmi_mar3 _8_bmi_mar3
_9_bmi_mar3 _10_bmi_mar3
save mar3_final_short.dta, replace

qui generate bias1=[(_1_bmi_mar3)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar3==.
qui generate bias2=[(_2_bmi_mar3)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar3==.
qui generate bias3=[(_3_bmi_mar3)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar3==.
qui generate bias4=[(_4_bmi_mar3)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar3==.
qui generate bias5=[(_5_bmi_mar3)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar3==.
qui generate bias6=[(_6_bmi_mar3)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar3==.
qui generate bias7=[(_7_bmi_mar3)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar3==.
qui generate bias8=[(_8_bmi_mar3)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar3==.
qui generate bias9=[(_9_bmi_mar3)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar3==.
qui generate bias10=[(_10_bmi_mar3)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar3==.
qui generate
mar3_bias_tot_1=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)+(bias8
)+(bias9)+(bias10)]/10
summ mar3_bias_tot_1, detail
ci means mar3_bias_tot_1

qui generate error_1=[(_1_bmi_mar3)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar3==.
qui generate error_2=[(_2_bmi_mar3)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar3==.
qui generate error_3=[(_3_bmi_mar3)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar3==.
qui generate error_4=[(_4_bmi_mar3)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar3==.
qui generate error_5=[(_5_bmi_mar3)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar3==.
qui generate error_6=[(_6_bmi_mar3)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar3==.
```

```
qui generate error_7=[(_7_bmi_mar3)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar3==.
qui generate error_8=[(_8_bmi_mar3)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar3==.
qui generate error_9=[(_9_bmi_mar3)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar3==.
qui generate error_10=[(_10_bmi_mar3)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar3==.

qui generate n_mse=[(error_1) + (error_2) + (error_3) + (error_4) + (error_5) +
(error_6) + (error_7) + (error_8) + (error_9) + (error_10)]/10
summ n_mse, detail
ci means n_mse

qui generate error1=[(_1_bmi_mar3)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar3==.
qui generate error2=[(_2_bmi_mar3)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar3==.
qui generate error3=[(_3_bmi_mar3)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar3==.
qui generate error4=[(_4_bmi_mar3)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar3==.
qui generate error5=[(_5_bmi_mar3)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar3==.
qui generate error6=[(_6_bmi_mar3)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar3==.
qui generate error7=[(_7_bmi_mar3)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar3==.
qui generate error8=[(_8_bmi_mar3)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar3==.
qui generate error9=[(_9_bmi_mar3)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar3==.
qui generate error10=[(_10_bmi_mar3)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar3==.

qui generate m_error=[(error1) + (error2) + (error3) + (error4) + (error5) +
(error6) + (error7) + (error8) + (error9) + (error10)]/10
summ m_error, detail
ci means m_error

qui generate mse=[m_error*m_error]
summ mse, detail
ci means mse


qui generate cc1=.
qui replace cc1=1 if _1_bmi_mar3<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc1=1 if _1_bmi_mar3>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc1=0 if _mi_miss==1 & cc1==. & bmi_mar3==.
qui generate cc2=.
qui replace cc2=1 if _2_bmi_mar3<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc2=1 if _2_bmi_mar3>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc2=0 if _mi_miss==1 & cc2==. & bmi_mar3==.
qui generate cc3=.
qui replace cc3=1 if _3_bmi_mar3<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar3==.
```

```
qui replace cc3=1 if _3_bmi_mar3>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc3=0 if _mi_miss==1 & cc3==. & bmi_mar3==.
qui generate cc4=.
qui replace cc4=1 if _4_bmi_mar3<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc4=1 if _4_bmi_mar3>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc4=0 if _mi_miss==1 & cc4==. & bmi_mar3==.
qui generate cc5=.
qui replace cc5=1 if _5_bmi_mar3<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc5=1 if _5_bmi_mar3>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc5=0 if _mi_miss==1 & cc5==. & bmi_mar3==.
qui generate cc6=.
qui replace cc6=1 if _6_bmi_mar3<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc6=1 if _6_bmi_mar3>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc6=0 if _mi_miss==1 & cc6==. & bmi_mar3==.
qui generate cc7=.
qui replace cc7=1 if _7_bmi_mar3<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc7=1 if _7_bmi_mar3>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc7=0 if _mi_miss==1 & cc7==. & bmi_mar3==.
qui generate cc8=.
qui replace cc8=1 if _8_bmi_mar3<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc8=1 if _8_bmi_mar3>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc8=0 if _mi_miss==1 & cc8==. & bmi_mar3==.
qui generate cc9=.
qui replace cc9=1 if _9_bmi_mar3<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc9=1 if _9_bmi_mar3>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc9=0 if _mi_miss==1 & cc9==. & bmi_mar3==.
qui generate cc10=.
qui replace cc10=1 if _10_bmi_mar3<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc10=1 if _10_bmi_mar3>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar3==.
qui replace cc10=0 if _mi_miss==1 & cc10==. & bmi_mar3==.
qui generate percent_cc=[cc1 + cc2 + cc3 + cc4 + cc5 + cc6 + cc7 + cc8 + cc9 +
cc10]/10
summ percent_cc, detail
ci means percent_cc

clear
cd "/Volumes/Thesis/High Dimensional/New MI Datasets"
  use mar4_final_dataset_1_mi.dta, clear
       foreach num of numlist 2/1000 {
          qui append using mar4_final_dataset_`num'_mi.dta
                }
save mar4_final_large.dta, replace
keep pracid patid bmi_mar4 calc_bmi_round _mi_miss _1_bmi_mar4 _2_bmi_mar4
_3_bmi_mar4 _4_bmi_mar4 _5_bmi_mar4 _6_bmi_mar4 _7_bmi_mar4 _8_bmi_mar4
_9_bmi_mar4 _10_bmi_mar4
```

```
save mar4_final_short.dta, replace

qui generate bias1=[(_1_bmi_mar4)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar4==.
qui generate bias2=[(_2_bmi_mar4)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar4==.
qui generate bias3=[(_3_bmi_mar4)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar4==.
qui generate bias4=[(_4_bmi_mar4)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar4==.
qui generate bias5=[(_5_bmi_mar4)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar4==.
qui generate bias6=[(_6_bmi_mar4)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar4==.
qui generate bias7=[(_7_bmi_mar4)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar4==.
qui generate bias8=[(_8_bmi_mar4)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar4==.
qui generate bias9=[(_9_bmi_mar4)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar4==.
qui generate bias10=[(_10_bmi_mar4)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mar4==.
qui generate
mar4_bias_tot_1=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)+(bias8
)+(bias9)+(bias10)]/10
summ mar4_bias_tot_1, detail
ci means mar4_bias_tot_1

qui generate error_1=[(_1_bmi_mar4)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar4==.
qui generate error_2=[(_2_bmi_mar4)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar4==.
qui generate error_3=[(_3_bmi_mar4)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar4==.
qui generate error_4=[(_4_bmi_mar4)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar4==.
qui generate error_5=[(_5_bmi_mar4)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar4==.
qui generate error_6=[(_6_bmi_mar4)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar4==.
qui generate error_7=[(_7_bmi_mar4)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar4==.
qui generate error_8=[(_8_bmi_mar4)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar4==.
qui generate error_9=[(_9_bmi_mar4)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar4==.
qui generate error_10=[(_10_bmi_mar4)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mar4==.

qui generate n_mse=[(error_1) + (error_2) + (error_3) + (error_4) + (error_5) +
(error_6) + (error_7) + (error_8) + (error_9) + (error_10)]/10
summ n_mse, detail
ci means n_mse

qui generate error1=[(_1_bmi_mar4)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar4==.
qui generate error2=[(_2_bmi_mar4)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar4==.
qui generate error3=[(_3_bmi_mar4)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar4==.
```

```
qui generate error4=[(_4_bmi_mar4)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar4==.
qui generate error5=[(_5_bmi_mar4)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar4==.
qui generate error6=[(_6_bmi_mar4)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar4==.
qui generate error7=[(_7_bmi_mar4)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar4==.
qui generate error8=[(_8_bmi_mar4)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar4==.
qui generate error9=[(_9_bmi_mar4)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar4==.
qui generate error10=[(_10_bmi_mar4)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mar4==.

qui generate m_error=[(error1) + (error2) + (error3) + (error4) + (error5) +
(error6) + (error7) + (error8) + (error9) + (error10)]/10
summ m_error, detail
ci means m_error

qui generate mse=[m_error*m_error]
summ mse, detail
ci means mse

qui generate cc1=.
qui replace cc1=1 if _1_bmi_mar4<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc1=1 if _1_bmi_mar4>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc1=0 if _mi_miss==1 & cc1==. & bmi_mar4==.
qui generate cc2=.
qui replace cc2=1 if _2_bmi_mar4<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc2=1 if _2_bmi_mar4>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc2=0 if _mi_miss==1 & cc2==. & bmi_mar4==.
qui generate cc3=.
qui replace cc3=1 if _3_bmi_mar4<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc3=1 if _3_bmi_mar4>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc3=0 if _mi_miss==1 & cc3==. & bmi_mar4==.
qui generate cc4=.
qui replace cc4=1 if _4_bmi_mar4<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc4=1 if _4_bmi_mar4>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc4=0 if _mi_miss==1 & cc4==. & bmi_mar4==.
qui generate cc5=.
qui replace cc5=1 if _5_bmi_mar4<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc5=1 if _5_bmi_mar4>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc5=0 if _mi_miss==1 & cc5==. & bmi_mar4==.
qui generate cc6=.
qui replace cc6=1 if _6_bmi_mar4<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc6=1 if _6_bmi_mar4>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc6=0 if _mi_miss==1 & cc6==. & bmi_mar4==.
```

```
qui generate cc7=.
qui replace cc7=1 if _7_bmi_mar4<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc7=1 if _7_bmi_mar4>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc7=0 if _mi_miss==1 & cc7==. & bmi_mar4==.
qui generate cc8=.
qui replace cc8=1 if _8_bmi_mar4<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc8=1 if _8_bmi_mar4>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc8=0 if _mi_miss==1 & cc8==. & bmi_mar4==.
qui generate cc9=.
qui replace cc9=1 if _9_bmi_mar4<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc9=1 if _9_bmi_mar4>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc9=0 if _mi_miss==1 & cc9==. & bmi_mar4==.
qui generate cc10=.
qui replace cc10=1 if _10_bmi_mar4<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc10=1 if _10_bmi_mar4>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mar4==.
qui replace cc10=0 if _mi_miss==1 & cc10==. & bmi_mar4==.
qui generate percent_cc=[cc1 + cc2 + cc3 + cc4 + cc5 + cc6 + cc7 + cc8 + cc9 +
cc10]/10
summ percent_cc, detail
ci means percent_cc

clear
cd "/Volumes/Thesis/High Dimensional/New MI Datasets"
  use mnar1_final_dataset_1_mi.dta, clear
       foreach num of numlist 2/1000 {
          qui append using mnar1_final_dataset_`num'_mi.dta
               }

save mnar1_final_large.dta, replace
keep pracid patid bmi_mnar1 calc_bmi_round _mi_miss _1_bmi_mnar1 _2_bmi_mnar1
_3_bmi_mnar1 _4_bmi_mnar1 _5_bmi_mnar1 _6_bmi_mnar1 _7_bmi_mnar1 _8_bmi_mnar1
_9_bmi_mnar1 _10_bmi_mnar1
save mnar1_final_short.dta, replace

qui generate bias1=[(_1_bmi_mnar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar1==.
qui generate bias2=[(_2_bmi_mnar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar1==.
qui generate bias3=[(_3_bmi_mnar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar1==.
qui generate bias4=[(_4_bmi_mnar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar1==.
qui generate bias5=[(_5_bmi_mnar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar1==.
qui generate bias6=[(_6_bmi_mnar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar1==.
qui generate bias7=[(_7_bmi_mnar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar1==.
qui generate bias8=[(_8_bmi_mnar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar1==.
qui generate bias9=[(_9_bmi_mnar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar1==.
```

```
qui generate bias10=[(_10_bmi_mnar1)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar1==.
qui generate
mnar1_bias_tot_1=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)+(bias
8)+(bias9)+(bias10)]/10
summ mnar1_bias_tot_1, detail
ci means mnar1_bias_tot_1

qui generate error_1=[(_1_bmi_mnar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar1==.
qui generate error_2=[(_2_bmi_mnar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar1==.
qui generate error_3=[(_3_bmi_mnar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar1==.
qui generate error_4=[(_4_bmi_mnar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar1==.
qui generate error_5=[(_5_bmi_mnar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar1==.
qui generate error_6=[(_6_bmi_mnar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar1==.
qui generate error_7=[(_7_bmi_mnar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar1==.
qui generate error_8=[(_8_bmi_mnar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar1==.
qui generate error_9=[(_9_bmi_mnar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar1==.
qui generate error_10=[(_10_bmi_mnar1)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar1==.

qui generate n_mse=[(error_1) + (error_2) + (error_3) + (error_4) + (error_5) +
(error_6) + (error_7) + (error_8) + (error_9) + (error_10)]/10
summ n_mse, detail
ci means n_mse

qui generate error1=[(_1_bmi_mnar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar1==.
qui generate error2=[(_2_bmi_mnar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar1==.
qui generate error3=[(_3_bmi_mnar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar1==.
qui generate error4=[(_4_bmi_mnar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar1==.
qui generate error5=[(_5_bmi_mnar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar1==.
qui generate error6=[(_6_bmi_mnar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar1==.
qui generate error7=[(_7_bmi_mnar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar1==.
qui generate error8=[(_8_bmi_mnar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar1==.
qui generate error9=[(_9_bmi_mnar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar1==.
qui generate error10=[(_10_bmi_mnar1)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar1==.

qui generate m_error=[(error1) + (error2) + (error3) + (error4) + (error5) +
(error6) + (error7) + (error8) + (error9) + (error10)]/10
summ m_error, detail
ci means m_error
```

```
qui generate mse=[m_error*m_error]
summ mse, detail
ci means mse

qui generate cc1=.
qui replace cc1=1 if _1_bmi_mnar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc1=1 if _1_bmi_mnar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc1=0 if _mi_miss==1 & cc1==. & bmi_mnar1==.
qui generate cc2=.
qui replace cc2=1 if _2_bmi_mnar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc2=1 if _2_bmi_mnar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc2=0 if _mi_miss==1 & cc2==. & bmi_mnar1==.
qui generate cc3=.
qui replace cc3=1 if _3_bmi_mnar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc3=1 if _3_bmi_mnar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc3=0 if _mi_miss==1 & cc3==. & bmi_mnar1==.
qui generate cc4=.
qui replace cc4=1 if _4_bmi_mnar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc4=1 if _4_bmi_mnar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc4=0 if _mi_miss==1 & cc4==. & bmi_mnar1==.
qui generate cc5=.
qui replace cc5=1 if _5_bmi_mnar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc5=1 if _5_bmi_mnar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc5=0 if _mi_miss==1 & cc5==. & bmi_mnar1==.
qui generate cc6=.
qui replace cc6=1 if _6_bmi_mnar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc6=1 if _6_bmi_mnar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc6=0 if _mi_miss==1 & cc6==. & bmi_mnar1==.
qui generate cc7=.
qui replace cc7=1 if _7_bmi_mnar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc7=1 if _7_bmi_mnar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc7=0 if _mi_miss==1 & cc7==. & bmi_mnar1==.
qui generate cc8=.
qui replace cc8=1 if _8_bmi_mnar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc8=1 if _8_bmi_mnar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc8=0 if _mi_miss==1 & cc8==. & bmi_mnar1==.
qui generate cc9=.
qui replace cc9=1 if _9_bmi_mnar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc9=1 if _9_bmi_mnar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc9=0 if _mi_miss==1 & cc9==. & bmi_mnar1==.
qui generate cc10=.
```

```
qui replace cc10=1 if _10_bmi_mnar1<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc10=1 if _10_bmi_mnar1>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar1==.
qui replace cc10=0 if _mi_miss==1 & cc10==. & bmi_mnar1==.
qui generate percent_cc=[cc1 + cc2 + cc3 + cc4 + cc5 + cc6 + cc7 + cc8 + cc9 +
cc10]/10
summ percent_cc, detail
ci means percent_cc


clear
cd "/Volumes/Thesis/High Dimensional/New MI Datasets"
  use mnar2_th_final_dataset_1_mi.dta, clear
       foreach num of numlist 2/1000 {
          qui append using mnar2_th_final_dataset_`num'_mi.dta
               }

save mnar2_th_final_large.dta, replace
keep pracid patid bmi_mnar2_th calc_bmi_round _mi_miss _1_bmi_mnar2_th
_2_bmi_mnar2_th _3_bmi_mnar2_th _4_bmi_mnar2_th _5_bmi_mnar2_th _6_bmi_mnar2_th
_7_bmi_mnar2_th _8_bmi_mnar2_th _9_bmi_mnar2_th _10_bmi_mnar2_th
save mnar2_th_final_short.dta, replace

qui generate bias1=[(_1_bmi_mnar2_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar2_th==.
qui generate bias2=[(_2_bmi_mnar2_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar2_th==.
qui generate bias3=[(_3_bmi_mnar2_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar2_th==.
qui generate bias4=[(_4_bmi_mnar2_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar2_th==.
qui generate bias5=[(_5_bmi_mnar2_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar2_th==.
qui generate bias6=[(_6_bmi_mnar2_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar2_th==.
qui generate bias7=[(_7_bmi_mnar2_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar2_th==.
qui generate bias8=[(_8_bmi_mnar2_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar2_th==.
qui generate bias9=[(_9_bmi_mnar2_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar2_th==.
qui generate bias10=[(_10_bmi_mnar2_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar2_th==.
qui generate
mnar2_th_bias_tot_1=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)+(b
ias8)+(bias9)+(bias10)]/10
summ mnar2_th_bias_tot_1, detail
ci means mnar2_th_bias_tot_1

qui generate error_1=[(_1_bmi_mnar2_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar2_th==.
qui generate error_2=[(_2_bmi_mnar2_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar2_th==.
qui generate error_3=[(_3_bmi_mnar2_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar2_th==.
qui generate error_4=[(_4_bmi_mnar2_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar2_th==.
qui generate error_5=[(_5_bmi_mnar2_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar2_th==.
```

```
qui generate error_6=[(_6_bmi_mnar2_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar2_th==.
qui generate error_7=[(_7_bmi_mnar2_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar2_th==.
qui generate error_8=[(_8_bmi_mnar2_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar2_th==.
qui generate error_9=[(_9_bmi_mnar2_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar2_th==.
qui generate error_10=[(_10_bmi_mnar2_th)-(calc_bmi_round)]^2/(calc_bmi_round)
if _mi_miss==1 & bmi_mnar2_th==.

qui generate n_mse=[(error_1) + (error_2) + (error_3) + (error_4) + (error_5) +
(error_6) + (error_7) + (error_8) + (error_9) + (error_10)]/10
summ n_mse, detail
ci means n_mse

qui generate error1=[(_1_bmi_mnar2_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar2_th==.
qui generate error2=[(_2_bmi_mnar2_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar2_th==.
qui generate error3=[(_3_bmi_mnar2_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar2_th==.
qui generate error4=[(_4_bmi_mnar2_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar2_th==.
qui generate error5=[(_5_bmi_mnar2_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar2_th==.
qui generate error6=[(_6_bmi_mnar2_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar2_th==.
qui generate error7=[(_7_bmi_mnar2_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar2_th==.
qui generate error8=[(_8_bmi_mnar2_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar2_th==.
qui generate error9=[(_9_bmi_mnar2_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar2_th==.
qui generate error10=[(_10_bmi_mnar2_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar2_th==.

qui generate m_error=[(error1) + (error2) + (error3) + (error4) + (error5) +
(error6) + (error7) + (error8) + (error9) + (error10)]/10
summ m_error, detail
ci means m_error

qui generate mse=[m_error*m_error]
summ mse, detail
ci means mse


qui generate cc1=.
qui replace cc1=1 if _1_bmi_mnar2_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc1=1 if _1_bmi_mnar2_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc1=0 if _mi_miss==1 & cc1==. & bmi_mnar2_th==.
qui generate cc2=.
qui replace cc2=1 if _2_bmi_mnar2_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc2=1 if _2_bmi_mnar2_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc2=0 if _mi_miss==1 & cc2==. & bmi_mnar2_th==.
qui generate cc3=.
```

```
qui replace cc3=1 if _3_bmi_mnar2_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc3=1 if _3_bmi_mnar2_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc3=0 if _mi_miss==1 & cc3==. & bmi_mnar2_th==.
qui generate cc4=.
qui replace cc4=1 if _4_bmi_mnar2_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc4=1 if _4_bmi_mnar2_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc4=0 if _mi_miss==1 & cc4==. & bmi_mnar2_th==.
qui generate cc5=.
qui replace cc5=1 if _5_bmi_mnar2_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc5=1 if _5_bmi_mnar2_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc5=0 if _mi_miss==1 & cc5==. & bmi_mnar2_th==.
qui generate cc6=.
qui replace cc6=1 if _6_bmi_mnar2_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc6=1 if _6_bmi_mnar2_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc6=0 if _mi_miss==1 & cc6==. & bmi_mnar2_th==.
qui generate cc7=.
qui replace cc7=1 if _7_bmi_mnar2_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc7=1 if _7_bmi_mnar2_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc7=0 if _mi_miss==1 & cc7==. & bmi_mnar2_th==.
qui generate cc8=.
qui replace cc8=1 if _8_bmi_mnar2_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc8=1 if _8_bmi_mnar2_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc8=0 if _mi_miss==1 & cc8==.  & bmi_mnar2_th==.
qui generate cc9=.
qui replace cc9=1 if _9_bmi_mnar2_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc9=1 if _9_bmi_mnar2_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc9=0 if _mi_miss==1 & cc9==.  & bmi_mnar2_th==.
qui generate cc10=.
qui replace cc10=1 if _10_bmi_mnar2_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc10=1 if _10_bmi_mnar2_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar2_th==.
qui replace cc10=0 if _mi_miss==1 & cc10==. & bmi_mnar2_th==.
qui generate percent_cc=[cc1 + cc2 + cc3 + cc4 + cc5 + cc6 + cc7 + cc8 + cc9 +
cc10]/10
summ percent_cc, detail
ci means percent_cc


clear
cd "/Volumes/Thesis/High Dimensional/New MI Datasets"
  use mnar3_th_final_dataset_1_mi.dta, clear
       foreach num of numlist 2/1000 {
          qui append using mnar3_th_final_dataset_`num'_mi.dta
                }
save mnar3_th_final_large.dta, replace
```

```
keep pracid patid bmi_mnar3_th calc_bmi_round _mi_miss _1_bmi_mnar3_th
_2_bmi_mnar3_th _3_bmi_mnar3_th _4_bmi_mnar3_th _5_bmi_mnar3_th _6_bmi_mnar3_th
_7_bmi_mnar3_th _8_bmi_mnar3_th _9_bmi_mnar3_th _10_bmi_mnar3_th
save mnar3_th_final_short.dta, replace

qui generate bias1=[(_1_bmi_mnar3_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar3_th==.
qui generate bias2=[(_2_bmi_mnar3_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar3_th==.
qui generate bias3=[(_3_bmi_mnar3_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar3_th==.
qui generate bias4=[(_4_bmi_mnar3_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar3_th==.
qui generate bias5=[(_5_bmi_mnar3_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar3_th==.
qui generate bias6=[(_6_bmi_mnar3_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar3_th==.
qui generate bias7=[(_7_bmi_mnar3_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar3_th==.
qui generate bias8=[(_8_bmi_mnar3_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar3_th==.
qui generate bias9=[(_9_bmi_mnar3_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar3_th==.
qui generate bias10=[(_10_bmi_mnar3_th)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar3_th==.
qui generate
mnar3_th_bias_tot_1=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)+(b
ias8)+(bias9)+(bias10)]/10
summ mnar3_th_bias_tot_1, detail
ci means mnar3_th_bias_tot_1

qui generate error_1=[(_1_bmi_mnar3_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar3_th==.
qui generate error_2=[(_2_bmi_mnar3_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar3_th==.
qui generate error_3=[(_3_bmi_mnar3_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar3_th==.
qui generate error_4=[(_4_bmi_mnar3_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar3_th==.
qui generate error_5=[(_5_bmi_mnar3_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar3_th==.
qui generate error_6=[(_6_bmi_mnar3_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar3_th==.
qui generate error_7=[(_7_bmi_mnar3_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar3_th==.
qui generate error_8=[(_8_bmi_mnar3_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar3_th==.
qui generate error_9=[(_9_bmi_mnar3_th)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar3_th==.
qui generate error_10=[(_10_bmi_mnar3_th)-(calc_bmi_round)]^2/(calc_bmi_round)
if _mi_miss==1 & bmi_mnar3_th==.

qui generate n_mse=[(error_1) + (error_2) + (error_3) + (error_4) + (error_5) +
(error_6) + (error_7) + (error_8) + (error_9) + (error_10)]/10
summ n_mse, detail
ci means n_mse

qui generate error1=[(_1_bmi_mnar3_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar3_th==.
```

```
qui generate error2=[(_2_bmi_mnar3_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar3_th==.
qui generate error3=[(_3_bmi_mnar3_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar3_th==.
qui generate error4=[(_4_bmi_mnar3_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar3_th==.
qui generate error5=[(_5_bmi_mnar3_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar3_th==.
qui generate error6=[(_6_bmi_mnar3_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar3_th==.
qui generate error7=[(_7_bmi_mnar3_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar3_th==.
qui generate error8=[(_8_bmi_mnar3_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar3_th==.
qui generate error9=[(_9_bmi_mnar3_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar3_th==.
qui generate error10=[(_10_bmi_mnar3_th)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar3_th==.

qui generate m_error=[(error1) + (error2) + (error3) + (error4) + (error5) +
(error6) + (error7) + (error8) + (error9) + (error10)]/10
summ m_error, detail
ci means m_error

qui generate mse=[m_error*m_error]
summ mse, detail
ci means mse

qui generate cc1=.
qui replace cc1=1 if _1_bmi_mnar3_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc1=1 if _1_bmi_mnar3_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc1=0 if _mi_miss==1 & cc1==. & bmi_mnar3_th==.
qui generate cc2=.
qui replace cc2=1 if _2_bmi_mnar3_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc2=1 if _2_bmi_mnar3_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc2=0 if _mi_miss==1 & cc2==. & bmi_mnar3_th==.
qui generate cc3=.
qui replace cc3=1 if _3_bmi_mnar3_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc3=1 if _3_bmi_mnar3_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc3=0 if _mi_miss==1 & cc3==. & bmi_mnar3_th==.
qui generate cc4=.
qui replace cc4=1 if _4_bmi_mnar3_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc4=1 if _4_bmi_mnar3_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc4=0 if _mi_miss==1 & cc4==. & bmi_mnar3_th==.
qui generate cc5=.
qui replace cc5=1 if _5_bmi_mnar3_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc5=1 if _5_bmi_mnar3_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc5=0 if _mi_miss==1 & cc5==. & bmi_mnar3_th==.
qui generate cc6=.
```

```
qui replace cc6=1 if _6_bmi_mnar3_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc6=1 if _6_bmi_mnar3_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc6=0 if _mi_miss==1 & cc6==. & bmi_mnar3_th==.
qui generate cc7=.
qui replace cc7=1 if _7_bmi_mnar3_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc7=1 if _7_bmi_mnar3_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc7=0 if _mi_miss==1 & cc7==. & bmi_mnar3_th==.
qui generate cc8=.
qui replace cc8=1 if _8_bmi_mnar3_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc8=1 if _8_bmi_mnar3_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc8=0 if _mi_miss==1 & cc8==. & bmi_mnar3_th==.
qui generate cc9=.
qui replace cc9=1 if _9_bmi_mnar3_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc9=1 if _9_bmi_mnar3_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc9=0 if _mi_miss==1 & cc9==. & bmi_mnar3_th==.
qui generate cc10=.
qui replace cc10=1 if _10_bmi_mnar3_th<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc10=1 if _10_bmi_mnar3_th>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar3_th==.
qui replace cc10=0 if _mi_miss==1 & cc10==. & bmi_mnar3_th==.
qui generate percent_cc=[cc1 + cc2 + cc3 + cc4 + cc5 + cc6 + cc7 + cc8 + cc9 +
cc10]/10
summ percent_cc, detail
ci means percent_cc

clear
cd "/Volumes/Thesis/High Dimensional/New MI Datasets"
  use mnar4_3w_final_dataset_1_mi.dta, clear
       foreach num of numlist 2/1000 {
         qui append using mnar4_3w_final_dataset_`num'_mi.dta
               }
save mnar4_3w_final_large.dta, replace
keep pracid patid bmi_mnar4_3w calc_bmi_round _mi_miss _1_bmi_mnar4_3w
_2_bmi_mnar4_3w _3_bmi_mnar4_3w _4_bmi_mnar4_3w _5_bmi_mnar4_3w _6_bmi_mnar4_3w
_7_bmi_mnar4_3w _8_bmi_mnar4_3w _9_bmi_mnar4_3w _10_bmi_mnar4_3w
save mnar4_3w_final_short.dta, replace

qui generate bias1=[(_1_bmi_mnar4_3w)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate bias2=[(_2_bmi_mnar4_3w)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate bias3=[(_3_bmi_mnar4_3w)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate bias4=[(_4_bmi_mnar4_3w)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate bias5=[(_5_bmi_mnar4_3w)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate bias6=[(_6_bmi_mnar4_3w)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate bias7=[(_7_bmi_mnar4_3w)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar4_3w==.
```

```
qui generate bias8=[(_8_bmi_mnar4_3w)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate bias9=[(_9_bmi_mnar4_3w)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate bias10=[(_10_bmi_mnar4_3w)-(calc_bmi_round)]/calc_bmi_round if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate
mnar4_3w_bias_tot_1=[(bias1)+(bias2)+(bias3)+(bias4)+(bias5)+(bias6)+(bias7)+(b
ias8)+(bias9)+(bias10)]/10
summ mnar4_3w_bias_tot_1, detail
ci means mnar4_3w_bias_tot_1

qui generate error_1=[(_1_bmi_mnar4_3w)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate error_2=[(_2_bmi_mnar4_3w)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate error_3=[(_3_bmi_mnar4_3w)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate error_4=[(_4_bmi_mnar4_3w)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate error_5=[(_5_bmi_mnar4_3w)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate error_6=[(_6_bmi_mnar4_3w)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate error_7=[(_7_bmi_mnar4_3w)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate error_8=[(_8_bmi_mnar4_3w)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate error_9=[(_9_bmi_mnar4_3w)-(calc_bmi_round)]^2/(calc_bmi_round) if
_mi_miss==1 & bmi_mnar4_3w==.
qui generate error_10=[(_10_bmi_mnar4_3w)-(calc_bmi_round)]^2/(calc_bmi_round)
if _mi_miss==1 & bmi_mnar4_3w==.

qui generate n_mse=[(error_1) + (error_2) + (error_3) + (error_4) + (error_5) +
(error_6) + (error_7) + (error_8) + (error_9) + (error_10)]/10
summ n_mse, detail
ci means n_mse

qui generate error1=[(_1_bmi_mnar4_3w)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar4_3w==.
qui generate error2=[(_2_bmi_mnar4_3w)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar4_3w==.
qui generate error3=[(_3_bmi_mnar4_3w)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar4_3w==.
qui generate error4=[(_4_bmi_mnar4_3w)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar4_3w==.
qui generate error5=[(_5_bmi_mnar4_3w)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar4_3w==.
qui generate error6=[(_6_bmi_mnar4_3w)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar4_3w==.
qui generate error7=[(_7_bmi_mnar4_3w)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar4_3w==.
qui generate error8=[(_8_bmi_mnar4_3w)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar4_3w==.
qui generate error9=[(_9_bmi_mnar4_3w)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar4_3w==.
qui generate error10=[(_10_bmi_mnar4_3w)-(calc_bmi_round)] if _mi_miss==1 &
bmi_mnar4_3w==.
```

```
qui generate m_error=[(error1) + (error2) + (error3) + (error4) + (error5) +
(error6) + (error7) + (error8) + (error9) + (error10)]/10
summ m_error, detail
ci means m_error

qui generate mse=[m_error*m_error]
summ mse, detail
ci means mse

qui generate cc1=.
qui replace cc1=1 if _1_bmi_mnar4_3w<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc1=1 if _1_bmi_mnar4_3w>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc1=0 if _mi_miss==1 & cc1==. & bmi_mnar4_3w==.
qui generate cc2=.
qui replace cc2=1 if _2_bmi_mnar4_3w<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc2=1 if _2_bmi_mnar4_3w>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc2=0 if _mi_miss==1 & cc2==. & bmi_mnar4_3w==.
qui generate cc3=.
qui replace cc3=1 if _3_bmi_mnar4_3w<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc3=1 if _3_bmi_mnar4_3w>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc3=0 if _mi_miss==1 & cc3==. & bmi_mnar4_3w==.
qui generate cc4=.
qui replace cc4=1 if _4_bmi_mnar4_3w<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc4=1 if _4_bmi_mnar4_3w>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc4=0 if _mi_miss==1 & cc4==. & bmi_mnar4_3w==.
qui generate cc5=.
qui replace cc5=1 if _5_bmi_mnar4_3w<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc5=1 if _5_bmi_mnar4_3w>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc5=0 if _mi_miss==1 & cc5==. & bmi_mnar4_3w==.
qui generate cc6=.
qui replace cc6=1 if _6_bmi_mnar4_3w<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc6=1 if _6_bmi_mnar4_3w>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc6=0 if _mi_miss==1 & cc6==. & bmi_mnar4_3w==.
qui generate cc7=.
qui replace cc7=1 if _7_bmi_mnar4_3w<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc7=1 if _7_bmi_mnar4_3w>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc7=0 if _mi_miss==1 & cc7==. & bmi_mnar4_3w==.
qui generate cc8=.
qui replace cc8=1 if _8_bmi_mnar4_3w<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc8=1 if _8_bmi_mnar4_3w>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc8=0 if _mi_miss==1 & cc8==. & bmi_mnar4_3w==.
qui generate cc9=.
qui replace cc9=1 if _9_bmi_mnar4_3w<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar4_3w==.
```

```
qui replace cc9=1 if _9_bmi_mnar4_3w>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc9=0 if _mi_miss==1 & cc9==. & bmi_mnar4_3w==.
qui generate cc10=.
qui replace cc10=1 if _10_bmi_mnar4_3w<30 & calc_bmi_round<30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc10=1 if _10_bmi_mnar4_3w>=30 & calc_bmi_round>=30 & _mi_miss==1 &
bmi_mnar4_3w==.
qui replace cc10=0 if _mi_miss==1 & cc10==. & bmi_mnar4_3w==.
qui generate percent_cc=[cc1 + cc2 + cc3 + cc4 + cc5 + cc6 + cc7 + cc8 + cc9 +
cc10]/10
summ percent_cc, detail
ci means percent_cc
```

Appendix Item 4.4: R Script for CART and Linear Regression Analyses

```
### Script for missing value imputation for BMI data
###
### Input files: code_bmi_dataset_demo.dta subsets
### Output files: runningResultSeed1.csv
###
### Imported libraries: haven, dpylr, mice, Hmisc, rpart, rpart.plot, randomForest

library(haven)
library(dplyr)
library(mice)
library(Hmisc)
library(rpart)
library(rpart.plot)
#library(randomForest)

### Initialize variables

running_denom_mcar = 0
running_denom_mar1 = 0
running_denom_mnar1 = 0

running_true_bmi_mcar = 0
running_true_bmi_mar1 = 0
running_true_bmi_mnar1 = 0

reg_bias_running_sum_mcar = 0
reg_bias_running_sum_mar1 = 0
reg_bias_running_sum_mnar1 = 0

CART_bias_running_sum_mcar = 0
CART_bias_running_sum_mar1 = 0
CART_bias_running_sum_mnar1 = 0

# rf_bias_running_sum_mcar = 0
# rf_bias_running_sum_mar1 = 0
# rf_bias_running_sum_mnar1 = 0

reg_sq_error_running_sum_mcar = 0
reg_sq_error_running_sum_mar1 = 0
reg_sq_error_running_sum_mnar1 = 0

CART_sq_error_running_sum_mcar = 0
CART_sq_error_running_sum_mar1 = 0
CART_sq_error_running_sum_mnar1 = 0

# rf_sq_error_running_sum_mcar = 0
# rf_sq_error_running_sum_mar1 = 0
# rf_sq_error_running_sum_mnar1 = 0

missing_types <- c("mcar", "mar1", "mnar1")

# setwd("~/Documents/Google Drive/Research/Machine learning")

for (seed_loop in 1:1) {

  for (iteration_loop in 1:1) {

    filename <-
paste("code_bmi_dataset/code_bmi_dataset_",seed_loop,"_",iteration_loop,".dta", sep = "")

    bmiData <- read_dta(filename)
    bmiData$pracid <- as.factor(bmiData$pracid)
    numObs <- nrow(bmiData)

    ## Reduce number of variables for toy model
```

```
    # bmiData <- bmiData[,c(1:90,5430:5440)]

    # numVars <- 100
    # demoVars <- 5430:5440
    # columns <- c(1:(numVars-11), demoVars)
    # bmiData <- bmiData[,columns]

    # Sort on bmi and generate a record number, then reorder
    bmiData <- arrange(bmiData, calc_bmi_round)
    bmiData$recno <- 1:numObs
    # reorderVars <- c(numVars+1,1:5,numVars,6:numVars)
    # bmiData <- bmiData[,reorderVars]

    # Create time variables
    bmiData$fuTime <- (bmiData$end_date - bmiData$start_date)/365.25
    bmiData$bmiTime <- (bmiData$bmi_date - bmiData$start_date)/365.25

    # Label missing bmi's
    bmiData$missing_mcar <- is.na(bmiData$bmi_mcar)
    bmiData[which(bmiData$missing_mcar == TRUE),"missing_mcar"] <- 1
    bmiData[which(bmiData$missing_mcar == FALSE),"missing_mcar"] <- 0

    bmiData$missing_mar1 <- is.na(bmiData$bmi_mar1)
    bmiData[which(bmiData$missing_mar1 == TRUE),"missing_mar1"] <- 1
    bmiData[which(bmiData$missing_mar1 == FALSE),"missing_mar1"] <- 0

    bmiData$missing_mnar1 <- is.na(bmiData$bmi_mnar1)
    bmiData[which(bmiData$missing_mnar1 == TRUE),"missing_mnar1"] <- 1
    bmiData[which(bmiData$missing_mnar1 == FALSE),"missing_mnar1"] <- 0

    # Create training and test sets (train on the non-missing for each type of missing
and test on the missing)

    trainSet_mcar <- bmiData[which(bmiData$missing_mcar == 0),]
    trainSet_mar1 <- bmiData[which(bmiData$missing_mar1 == 0),]
    trainSet_mnar1 <- bmiData[which(bmiData$missing_mnar1 == 0),]

    testSet_mcar <- bmiData[which(bmiData$missing_mcar == 1),]
    testSet_mar1 <- bmiData[which(bmiData$missing_mar1 == 1),]
    testSet_mnar1 <- bmiData[which(bmiData$missing_mnar1 == 1),]

    # Subset only analyzable variables

    trainAnalysisSet_mcar <- select(trainSet_mcar, calc_bmi_round, -
starts_with("missing"), -pracid, -patid, starts_with("code"), age:death,
+contains("Time"), -start_date, -end_date, -transfer_date, -death_date, -
starts_with("bmi_m"), -urbrural, -townsend)
    trainAnalysisSet_mar1 <- select(trainSet_mar1, calc_bmi_round, -
starts_with("missing"), -pracid, -patid, starts_with("code"), age:death,
+contains("Time"), -start_date, -end_date, -transfer_date, -death_date, -
starts_with("bmi_m"), -urbrural, -townsend)
    trainAnalysisSet_mnar1 <- select(trainSet_mnar1, calc_bmi_round, -
starts_with("missing"), -pracid, -patid, starts_with("code"), age:death,
+contains("Time"), -start_date, -end_date, -transfer_date, -death_date, -
starts_with("bmi_m"), -urbrural, -townsend)

    testAnalysisSet_mcar <- select(testSet_mcar, calc_bmi_round, -starts_with("missing"),
-pracid, -patid, starts_with("code"), age:death, +contains("Time"), -start_date, -
end_date, -transfer_date, -death_date, -starts_with("bmi_m"), -urbrural, -townsend)
    testAnalysisSet_mar1 <- select(testSet_mar1, calc_bmi_round, -starts_with("missing"),
-pracid, -patid, starts_with("code"), age:death, +contains("Time"), -start_date, -
end_date, -transfer_date, -death_date, -starts_with("bmi_m"), -urbrural, -townsend)
    testAnalysisSet_mnar1 <- select(testSet_mnar1, calc_bmi_round, -
starts_with("missing"), -pracid, -patid, starts_with("code"), age:death,
+contains("Time"), -start_date, -end_date, -transfer_date, -death_date, -
starts_with("bmi_m"), -urbrural, -townsend)

    # RUN ANALYSIS
```

```
    for (type in missing_types) {

      trainSet <- paste("trainAnalysisSet_", type, sep = "")
      testSet <- paste("testAnalysisSet_", type, sep = "")

      denom <- paste("running_denom_", type, sep = "")
      eval(call("<-", as.name(denom), get(denom) + nrow(get(testSet)))) ## increments
denom

      true_bmi_run <- paste("running_true_bmi_", type, sep = "")
      eval(call("<-", as.name(true_bmi_run), get(true_bmi_run) +
sum(get(testSet)$calc_bmi_round)))

      # Linear regression

      reg_bias_run <- paste("reg_bias_running_sum_", type, sep = "")
      reg_sq_error_run <- paste("reg_sq_error_running_sum_", type, sep = "")

      regressBMI_fit <- lm(calc_bmi_round ~ ., data = get(trainSet))
      BMI_regress_prediction <- predict(regressBMI_fit, get(testSet))
      errorRegress <- (BMI_regress_prediction - get(testSet)$calc_bmi_round)
      biasRegress <-  errorRegress/get(testSet)$calc_bmi_round
      sq_error_regress <- errorRegress^2

      eval(call("<-", as.name(reg_bias_run), get(reg_bias_run) + sum(biasRegress)))
      eval(call("<-", as.name(reg_sq_error_run), get(reg_sq_error_run) +
sum(sq_error_regress)))

      # CART

      CART_bias_run <- paste("CART_bias_running_sum_", type, sep = "")
      CART_sq_error_run <- paste("CART_sq_error_running_sum_", type, sep = "")

      CART_BMI_fit <- rpart(calc_bmi_round ~ ., data = get(trainSet), method = "anova")
      BMI_CART_prediction <- predict(CART_BMI_fit, get(testSet), type = "vector")
      errorCART <- (BMI_CART_prediction - get(testSet)$calc_bmi_round)
      biasCART <-  errorCART/get(testSet)$calc_bmi_round
      sq_error_CART <- errorCART^2

      eval(call("<-", as.name(CART_bias_run), get(CART_bias_run) + sum(biasCART)))
      eval(call("<-", as.name(CART_sq_error_run), get(CART_sq_error_run) +
sum(sq_error_CART)))

      # # Random forest
      #
      # rf_bias_run <- paste("rf_bias_running_sum_", type, sep = "")
      # rf_sq_error_run <- paste("rf_sq_error_running_sum_", type, sep = "")
      #
      # rfBMI_fit <- randomForest(calc_bmi_round ~ ., data = get(trainSet), importance =
TRUE, ntree = 1000, na.action = na.roughfix)
      # BMI_rf_prediction <- predict(rfBMI_fit, get(testSet))
      # errorRF <- (BMI_rf_prediction - get(testSet)$calc_bmi_round)
      # biasRF <-  errorRF/get(testSet)$calc_bmi_round
      # sq_error_RF <- errorRF^2
      #
      # eval(call("<-", as.name(rf_bias_run), get(rf_bias_run) + sum(biasRF)))
      # eval(call("<-", as.name(rf_sq_error_run), get(rf_sq_error_run) +
sum(sq_error_RF)))
      #
      # if (seed_loop == 4 & iteration_loop == 1) {
      #   impPlot_type <- paste("impPlot_",type,"4SHORT.jpg", sep = "")
      #   jpeg(impPlot_type)
      #   varImpPlot(rfBMI_fit, n.var = 10, main = type)
      #   dev.off()
      # }

    }
```

```
  } # close iteration_loop

} # close seed_loop


# CALCULATE AND ORGANIZE FINAL RESULTS

# bias_reg_mcar = reg_bias_running_sum_mcar/running_denom_mcar
# bias_reg_mar1 = reg_bias_running_sum_mar1/running_denom_mar1
# bias_reg_mnar1 = reg_bias_running_sum_mnar1/running_denom_mnar1
#
# MSE_reg_mcar =
(reg_sq_error_running_sum_mcar/running_denom_mcar)/(running_true_bmi_mcar/running_denom_m
car)
# MSE_reg_mar1 =
(reg_sq_error_running_sum_mar1/running_denom_mar1)/(running_true_bmi_mar1/running_denom_m
ar1)
# MSE_reg_mnar1 =
(reg_sq_error_running_sum_mnar1/running_denom_mnar1)/(running_true_bmi_mnar1/running_deno
m_mnar1)
#
# bias_CART_mcar = CART_bias_running_sum_mcar/running_denom_mcar
# bias_CART_mar1 = CART_bias_running_sum_mar1/running_denom_mar1
# bias_CART_mnar1 = CART_bias_running_sum_mnar1/running_denom_mnar1
#
# MSE_CART_mcar =
(CART_sq_error_running_sum_mcar/running_denom_mcar)/(running_true_bmi_mcar/running_denom_
mcar)
# MSE_CART_mar1 =
(CART_sq_error_running_sum_mar1/running_denom_mar1)/(running_true_bmi_mar1/running_denom_
mar1)
# MSE_CART_mnar1 =
(CART_sq_error_running_sum_mnar1/running_denom_mnar1)/(running_true_bmi_mnar1/running_den
om_mnar1)

# bias_rf_mcar = rf_bias_running_sum_mcar/running_denom_mcar
# bias_rf_mar1 = rf_bias_running_sum_mar1/running_denom_mar1
# bias_rf_mnar1 = rf_bias_running_sum_mnar1/running_denom_mnar1

# MSE_rf_mcar =
(rf_sq_error_running_sum_mcar/running_denom_mcar)/(running_true_bmi_mcar/running_denom_mc
ar)
# MSE_rf_mar1 =
(rf_sq_error_running_sum_mar1/running_denom_mar1)/(running_true_bmi_mar1/running_denom_ma
r1)
# MSE_rf_mnar1 =
(rf_sq_error_running_sum_mnar1/running_denom_mnar1)/(running_true_bmi_mnar1/running_denom
_mnar1)

runningResult <- data.frame(matrix(nrow = 3, ncol = 7))
colnames(runningResult) <- c("type", "reg_bias", "reg_sq_error", "CART_bias",
"CART_sq_error", "running_denom", "running_true_bmi")

runningResult[1,1] <- "mcar"
runningResult[2,1] <- "mar1"
runningResult[3,1] <- "mnar1"

runningResult[1,2] <- reg_bias_running_sum_mcar
runningResult[2,2] <- reg_bias_running_sum_mar1
runningResult[3,2] <- reg_bias_running_sum_mnar1

runningResult[1,3] <- reg_sq_error_running_sum_mcar
runningResult[2,3] <- reg_sq_error_running_sum_mar1
runningResult[3,3] <- reg_sq_error_running_sum_mnar1

runningResult[1,4] <- CART_bias_running_sum_mcar
runningResult[2,4] <- CART_bias_running_sum_mar1
```

```
runningResult[3,4] <- CART_bias_running_sum_mnar1

runningResult[1,5] <- CART_sq_error_running_sum_mcar
runningResult[2,5] <- CART_sq_error_running_sum_mar1
runningResult[3,5] <- CART_sq_error_running_sum_mnar1

runningResult[1,6] <- running_denom_mcar
runningResult[2,6] <- running_denom_mar1
runningResult[3,6] <- running_denom_mnar1

runningResult[1,7] <- running_true_bmi_mcar
runningResult[2,7] <- running_true_bmi_mar1
runningResult[3,7] <- running_true_bmi_mnar1

write.table(runningResult, "runningResultSeed1.csv", col.names = TRUE)
```

BIBLIOGRAPHY

1. Davis GL, Alter MJ, El-Serag H, Poynard T, Jennings LW. Aging of hepatitis C virus (HCV)-infected persons in the United States: a multiple cohort model of HCV prevalence and disease progression. Gastroenterology 2010;138:513-21, 21 e1-6.

2. Ly KN, Xing J, Klevens RM, Jiles RB, Ward JW, Holmberg SD. The increasing burden of mortality from viral hepatitis in the United States between 1999 and 2007. Ann Intern Med 2012;156:271-8.

3. van der Meer AJ, Veldt BJ, Feld JJ, et al. Association between sustained virological response and all-cause mortality among patients with chronic hepatitis C and advanced hepatic fibrosis. JAMA 2012;308:2584-93.

4. Razavi H, Elkhoury AC, Elbasha E, et al. Chronic hepatitis C virus (HCV) disease burden and cost in the United States. Hepatology 2013;57:2164-70.

5. HCV Guidance: Recommendations for Testing, Managing, and Treating Hepatitis C. (Accessed August 1, 2017, 2017, at http://www.hcvguidelines.org.)

6. Polaris Observatory HCVC. Global prevalence and genotype distribution of hepatitis C virus infection in 2015: a modelling study. Lancet Gastroenterol Hepatol 2017;2:161-76.

7. Denniston MM, Jiles RB, Drobeniuc J, et al. Chronic hepatitis C virus infection in the United States, National Health and Nutrition Examination Survey 2003 to 2010. Ann Intern Med 2014;160:293-300.

8. Armstrong GL, Wasley A, Simard EP, McQuillan GM, Kuhnert WL, Alter MJ. The prevalence of hepatitis C virus infection in the United States, 1999 through 2002. Ann Intern Med 2006;144:705-14.

9. Alter MJ, Kruszon-Moran D, Nainan OV, et al. The prevalence of hepatitis C virus infection in the United States, 1988 through 1994. N Engl J Med 1999;341:556-62.

10. Bukh J, Purcell RH, Miller RH. At least 12 genotypes of hepatitis C virus predicted by sequence analysis of the putative E1 gene of isolates collected worldwide. Proc Natl Acad Sci U S A 1993;90:8234-8.

11. Ogata N, Alter HJ, Miller RH, Purcell RH. Nucleotide sequence and mutation rate of the H strain of hepatitis C virus. Proc Natl Acad Sci U S A 1991;88:3392-6.

12. Simmonds P, Holmes EC, Cha TA, et al. Classification of hepatitis C virus into six major genotypes and a series of subtypes by phylogenetic analysis of the NS-5 region. J Gen Virol 1993;74 (Pt 11):2391-9.

13. Bukh J, Miller RH, Purcell RH. Genetic heterogeneity of hepatitis C virus: quasispecies and genotypes. Semin Liver Dis 1995;15:41-63.

14. Smith DB, Bukh J, Kuiken C, et al. Expanded classification of hepatitis C virus into 7 genotypes and 67 subtypes: updated criteria and genotype assignment web resource. Hepatology 2014;59:318-27.

15. Ball JK, Tarr AW, McKeating JA. The past, present and future of neutralizing antibodies for hepatitis C virus. Antiviral Res 2014;105:100-11.

16. Holz L, Rehermann B. T cell responses in hepatitis C virus infection: historical overview and goals for future research. Antiviral Res 2015;114:96-105.

17. Seeff LB. Natural history of chronic hepatitis C. Hepatology 2002;36:S35-46.

18. Denniston MM, Klevens RM, McQuillan GM, Jiles RB. Awareness of infection, knowledge of hepatitis C, and medical follow-up among individuals testing positive for hepatitis C: National Health and Nutrition Examination Survey 2001-2008. Hepatology 2012;55:1652-61.

19. White DL, Thrift AP, Kanwal F, Davila J, El-Serag HB. Incidence of Hepatocellular Carcinoma in All 50 United States, From 2000 Through 2012. Gastroenterology 2017;152:812-20 e5.

20. El-Serag HB, Kanwal F. Epidemiology of hepatocellular carcinoma in the United States: where are we? Where do we go? Hepatology 2014;60:1767-75.

21. Kanwal F, Hoang T, Kramer JR, et al. Increasing prevalence of HCC and cirrhosis in patients with chronic hepatitis C virus infection. Gastroenterology 2011;140:1182-8 e1.

22. Lo Re V, 3rd, Gowda C, Urick PN, et al. Disparities in Absolute Denial of Modern Hepatitis C Therapy by Type of Insurance. Clin Gastroenterol Hepatol 2016;14:1035-43.

23. Chhatwal J, Wang X, Ayer T, et al. Hepatitis C Disease Burden in the United States in the era of oral direct-acting antivirals. Hepatology 2016;64:1442-50.

24. Stepanova M, Younossi ZM. Economic Burden of Hepatitis C Infection. Clin Liver Dis 2017;21:579-94.

25. Ireton RC, Gale M, Jr. Pushing to a cure by harnessing innate immunity against hepatitis C virus. Antiviral Res 2014;108:156-64.

26. Hiet MS, Bauhofer O, Zayas M, et al. Control of temporal activation of hepatitis C virus-induced interferon response by domain 2 of nonstructural protein 5A. J Hepatol 2015;63:829-37.

27. Shi J, Li Y, Chang W, Zhang X, Wang FS. Current progress in host innate and adaptive immunity against hepatitis C virus infection. Hepatol Int 2017;11:374-83.

28. Kramer B, Korner C, Kebschull M, et al. Natural killer p46High expression defines a natural killer cell subset that is potentially involved in control of hepatitis C virus replication and modulation of liver fibrosis. Hepatology 2012;56:1201-13.

29. Pembroke T, Christian A, Jones E, et al. The paradox of NKp46+ natural killer cells: drivers of severe hepatitis C virus-induced pathology but in-vivo resistance to interferon alpha treatment. Gut 2014;63:515-24.

30. Penna A, Pilli M, Zerbini A, et al. Dysfunction and functional restoration of HCV-specific CD8 responses in chronic hepatitis C virus infection. Hepatology 2007;45:588-601.

31. Lauer GM. Immune responses to hepatitis C virus (HCV) infection and the prospects for an effective HCV vaccine or immunotherapies. J Infect Dis 2013;207 Suppl 1:S7-S12.

32. Gumber SC, Chopra S. Hepatitis C: a multifaceted disease. Review of extrahepatic manifestations. Annals of internal medicine 1995;123:615-20.

33. Ferri C, Colaci M, Fallahi P, Ferrari SM, Antonelli A, Giuggioli D. Thyroid Involvement in Hepatitis C Virus-Infected Patients with/without Mixed Cryoglobulinemia. Front Endocrinol (Lausanne) 2017;8:159.

34. Younossi Z, Park H, Henry L, Adeyemi A, Stepanova M. Extrahepatic Manifestations of Hepatitis C: A Meta-analysis of Prevalence, Quality of Life, and Economic Burden. Gastroenterology 2016;150:1599-608.

35. Negro F, Forton D, Craxi A, Sulkowski MS, Feld JJ, Manns MP. Extrahepatic morbidity and mortality of chronic hepatitis C. Gastroenterology 2015;149:1345-60.

36. Mehta SH, Brancati FL, Sulkowski MS, Strathdee SA, Szklo M, Thomas DL. Prevalence of type 2 diabetes mellitus among persons with hepatitis C virus infection in the United States. Ann Intern Med 2000;133:592-9.

37. Mehta SH, Brancati FL, Strathdee SA, et al. Hepatitis C virus infection and incident type 2 diabetes. Hepatology 2003;38:50-6.

38. Taura N, Ichikawa T, Hamasaki K, et al. Association between liver fibrosis and insulin sensitivity in chronic hepatitis C patients. Am J Gastroenterol 2006;101:2752-9.

39. Wang Q, Chen J, Wang Y, Han X, Chen X. Hepatitis C virus induced a novel apoptosis-like death of pancreatic beta cells through a caspase 3-dependent pathway. PLoS One 2012;7:e38522.

40. Laskus T, Radkowski M, Wang LF, Vargas H, Rakela J. Search for hepatitis C virus extrahepatic replication sites in patients with acquired immunodeficiency syndrome: specific detection of negative-strand viral RNA in various tissues. Hepatology 1998;28:1398-401.

41. Betterle C, Fabris P, Zanchetta R, et al. Autoimmunity against pancreatic islets and other tissues before and after interferon-alpha therapy in patients with hepatitis C virus chronic infection. Diabetes Care 2000;23:1177-81.

42. Agnello V, Abel G, Elfahal M, Knight GB, Zhang QX. Hepatitis C virus and other flaviviridae viruses enter cells via low density lipoprotein receptor. Proc Natl Acad Sci U S A 1999;96:12766-71.

43. Wunschmann S, Medh JD, Klinzmann D, Schmidt WN, Stapleton JT. Characterization of hepatitis C virus (HCV) and HCV E2 interactions with CD81 and the low-density lipoprotein receptor. J Virol 2000;74:10055-62.

44. Andre P, Komurian-Pradel F, Deforges S, et al. Characterization of low- and very-low-density hepatitis C virus RNA-containing particles. J Virol 2002;76:6919-28.

45. Kapadia SB, Barth H, Baumert T, McKeating JA, Chisari FV. Initiation of hepatitis C virus infection is dependent on cholesterol and cooperativity between CD81 and scavenger receptor B type I. J Virol 2007;81:374-83.

46. Hishiki T, Shimizu Y, Tobita R, et al. Infectivity of hepatitis C virus is influenced by association with apolipoprotein E isoforms. J Virol 2010;84:12048-57.

47. Zhang FL, Casey PJ. Protein prenylation: molecular mechanisms and functional consequences. Annu Rev Biochem 1996;65:241-69.

48. Kapadia SB, Chisari FV. Hepatitis C virus RNA replication is regulated by host geranylgeranylation and fatty acids. Proc Natl Acad Sci U S A 2005;102:2561-6.

49. Su AI, Pezacki JP, Wodicka L, et al. Genomic analysis of the host response to hepatitis C virus infection. Proc Natl Acad Sci U S A 2002;99:15669-74.

50. Ye J, Wang C, Sumpter R, Jr., Brown MS, Goldstein JL, Gale M, Jr. Disruption of hepatitis C virus RNA replication through inhibition of host protein geranylgeranylation. Proc Natl Acad Sci U S A 2003;100:15865-70.

51. Dai CY, Yeh ML, Huang CF, et al. Chronic hepatitis C infection is associated with insulin resistance and lipid profiles. J Gastroenterol Hepatol 2015;30:879-84.

52. Lonardo A, Loria P, Adinolfi LE, Carulli N, Ruggiero G. Hepatitis C and steatosis: a reappraisal. J Viral Hepat 2006;13:73-80.

53. Serfaty L. Metabolic Manifestations of Hepatitis C Virus: Diabetes Mellitus, Dyslipidemia. Clin Liver Dis 2017;21:475-86.

54. Perlemuter G, Sabile A, Letteron P, et al. Hepatitis C virus core protein inhibits microsomal triglyceride transfer protein activity and very low density lipoprotein secretion: a model of viral-related steatosis. FASEB J 2002;16:185-94.

55. Dharancy S, Malapel M, Perlemuter G, et al. Impaired expression of the peroxisome proliferator-activated receptor alpha during hepatitis C virus infection. Gastroenterology 2005;128:334-42.

56. Athyros VG, Tziomalos K, Katsiki N, Doumas M, Karagiannis A, Mikhailidis DP. Cardiovascular risk across the histological spectrum and the clinical manifestations of non-alcoholic fatty liver disease: An update. World J Gastroenterol 2015;21:6820-34.

57. Stary HC, Chandler AB, Glagov S, et al. A definition of initial, fatty streak, and intermediate lesions of atherosclerosis. A report from the Committee on Vascular Lesions of the Council on Arteriosclerosis, American Heart Association. Circulation 1994;89:2462-78.

58. Libby P, Egan D, Skarlatos S. Roles of infectious agents in atherosclerosis and restenosis: an assessment of the evidence and need for future research. Circulation 1997;96:4095-103.

59. Danesh J, Collins R, Peto R. Chronic infections and coronary heart disease: is there a link? Lancet 1997;350:430-6.

60. Gershon AS, Margulies M, Gorczynski RM, Heathcote EJ. Serum cytokine values and fatigue in chronic hepatitis C infection. Journal of viral hepatitis 2000;7:397-402.

61. Momiyama Y, Ohmori R, Kato R, Taniguchi H, Nakamura H, Ohsuzu F. Lack of any association between persistent hepatitis B or C virus infection and coronary artery disease. Atherosclerosis 2005;181:211-3.

62. Vassalle C, Masini S, Bianchi F, Zucchelli GC. Evidence for association between hepatitis C virus seropositivity and coronary artery disease. Heart 2004;90:565-6.

63. Arcari CM, Nelson KE, Netski DM, Nieto FJ, Gaydos CA. No association between hepatitis C virus seropositivity and acute myocardial infarction. Clin Infect Dis 2006;43:e53-6.

64. Ishizaka N, Ishizaka Y, Takahashi E, et al. Association between hepatitis C virus seropositivity, carotid-artery plaque, and intima-media thickening. Lancet 2002;359:133-5.

65. Ishizaka Y, Ishizaka N, Takahashi E, et al. Association between hepatitis C virus core protein and carotid atherosclerosis. Circ J 2003;67:26-30.

66. Kiechl S, Egger G, Mayr M, et al. Chronic infections and the risk of carotid atherosclerosis: prospective results from a large population study. Circulation 2001;103:1064-70.

67. Bilora F, Campagnolo E, Rinaldi R, Rossato A, Arzenton M, Petrobelli F. Carotid and femoral atherosclerosis in chronic hepatitis C: a 5-year follow-up. Angiology 2008;59:717-20.

68. Bilora F, Rinaldi R, Boccioletti V, Petrobelli F, Girolami A. Chronic viral hepatitis: a prospective factor against atherosclerosis. A study with echo-color Doppler of the carotid and femoral arteries and the abdominal aorta. Gastroenterol Clin Biol 2002;26:1001-4.

69. Hyams KC, Smith TC, Riddle J, Trump DH, Gray G. Viral hepatitis in the U.S. military: a study of hospitalization records from 1974 to 1999. Mil Med 2001;166:862-5.

70. Hyams KC, Riddle J, Rubertone M, et al. Prevalence and incidence of hepatitis C virus infection in the US military: a seroepidemiologic survey of 21,000 troops. Am J Epidemiol 2001;153:764-70.

71. Kramer JR, Kanwal F, Richardson P, Giordano TP, Petersen LA, El-Serag HB. Importance of patient, provider, and facility predictors of hepatitis C virus treatment in veterans: a national study. Am J Gastroenterol 2011;106:483-91.

72. Edlin BR, Eckhardt BJ, Shu MA, Holmberg SD, Swan T. Toward a more accurate estimate of the prevalence of hepatitis C in the United States. Hepatology 2015;62:1353-63.

73. Do A, Mittal Y, Liapakis A, et al. Drug Authorization for Sofosbuvir/Ledipasvir (Harvoni) for Chronic HCV Infection in a Real-World Cohort: A New Barrier in the HCV Care Cascade. PLoS One 2015;10:e0135645.

74. Thomssen R, Bonk S, Propfe C, Heermann KH, Kochel HG, Uy A. Association of hepatitis C virus in human sera with beta-lipoprotein. Med Microbiol Immunol 1992;181:293-300.

75. Thomssen R, Bonk S, Thiele A. Density heterogeneities of hepatitis C virus in human sera due to the binding of beta-lipoproteins and immunoglobulins. Med Microbiol Immunol 1993;182:329-34.

76. Ikeda M, Abe K, Yamada M, Dansako H, Naka K, Kato N. Different anti-HCV profiles of statins and their potential for combination therapy with interferon. Hepatology 2006;44:117-25.

77. Lo Re V, 3rd, Lim JK, Goetz MB, et al. Validity of diagnostic codes and liver-related laboratory abnormalities to identify hepatic decompensation events in the Veterans Aging Cohort Study. Pharmacoepidemiol Drug Saf 2011;20:689-99.

78. Stone NJ, Robinson JG, Lichtenstein AH, et al. 2013 ACC/AHA guideline on the treatment of blood cholesterol to reduce atherosclerotic cardiovascular risk in adults: a report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines. Circulation 2014;129:S1-45.

79. Vallianou NG, Kostantinou A, Kougias M, Kazazis C. Statins and cancer. Anticancer Agents Med Chem 2014;14:706-12.

80. National Heart L, Blood Institute ACTN, Truwit JD, et al. Rosuvastatin for sepsis-associated acute respiratory distress syndrome. N Engl J Med 2014;370:2191-200.

81. Kruger P, Bailey M, Bellomo R, et al. A multicenter randomized trial of atorvastatin therapy in intensive care patients with severe sepsis. Am J Respir Crit Care Med 2013;187:743-50.

82. Ikdahl E, Rollefstad S, Hisdal J, et al. Sustained Improvement of Arterial Stiffness and Blood Pressure after Long-Term Rosuvastatin Treatment in Patients with

Inflammatory Joint Diseases: Results from the RORA-AS Study. PLoS One 2016;11:e0153440.

83. Fatemi A, Moosavi M, Sayedbonakdar Z, Farajzadegan Z, Kazemi M, Smiley A. Atorvastatin effect on systemic lupus erythematosus disease activity: a double-blind randomized clinical trial. Clin Rheumatol 2014;33:1273-8.

84. Abraldes JG, Albillos A, Banares R, et al. Simvastatin lowers portal pressure in patients with cirrhosis and portal hypertension: a randomized controlled trial. Gastroenterology 2009;136:1651-8.

85. Abraldes JG, Villanueva C, Aracil C, et al. Addition of Simvastatin to Standard Therapy for the Prevention of Variceal Rebleeding Does Not Reduce Rebleeding but Increases Survival in Patients With Cirrhosis. Gastroenterology 2016;150:1160-70 e3.

86. Mohanty A, Tate JP, Garcia-Tsao G. Statins Are Associated With a Decreased Risk of Decompensation and Death in Veterans With Hepatitis C-Related Compensated Cirrhosis. Gastroenterology 2016;150:430-40 e1.

87. Lewis JH, Mortensen ME, Zweig S, et al. Efficacy and safety of high-dose pravastatin in hypercholesterolemic patients with well-compensated chronic liver disease: Results of a prospective, randomized, double-blind, placebo-controlled, multicenter trial. Hepatology 2007;46:1453-63.

88. Zhang T, Li Y, Lai JP, et al. Alcohol potentiates hepatitis C virus replicon expression. Hepatology 2003;38:57-65.

89. Weiss RD. Adherence to pharmacotherapy in patients with alcohol and opioid dependence. Addiction 2004;99:1382-92.

90. Boscarino JA, Moorman AC, Rupp LB, et al. Comparison of ICD-9 Codes for Depression and Alcohol Misuse to Survey Instruments Suggests These Codes Should Be Used with Caution. Dig Dis Sci 2017;62:2704-12.

91. Bush K, Kivlahan DR, McDonell MB, Fihn SD, Bradley KA. The AUDIT alcohol consumption questions (AUDIT-C): an effective brief screening test for problem drinking. Ambulatory Care Quality Improvement Project (ACQUIP). Alcohol Use Disorders Identification Test. Arch Intern Med 1998;158:1789-95.

92. Piette JD, Heisler M. Problems due to medication costs among VA and non-VA patients with chronic illnesses. Am J Manag Care 2004;10:861-8.

93. Pandya P, Rzouq F, Oni O. Sustained virologic response and other potential genotype-specific roles of statins among patients with hepatitis C-related chronic liver diseases. Clin Res Hepatol Gastroenterol 2015;39:555-65.

94. Atsukawa M, Tsubota A, Shimada N, et al. Effect of fluvastatin on 24-week telaprevir-based combination therapy for hepatitis C virus genotype 1b-infected chronic hepatitis C. Eur J Gastroenterol Hepatol 2014;26:781-7.

95. Selic Kurincic T, Lesnicar G, Poljak M, et al. Impact of added fluvastatin to standard-of-care treatment on sustained virological response in naive chronic hepatitis C Patients infected with genotypes 1 and 3. Intervirology 2014;57:23-30.

96. Bader T, Hughes LD, Fazili J, et al. A randomized controlled trial adding fluvastatin to peginterferon and ribavirin for naive genotype 1 hepatitis C patients. J Viral Hepat 2013;20:622-7.

97. Balogun MA, Ramsay ME, Hesketh LM, et al. The prevalence of hepatitis C in England and Wales. J Infect 2002;45:219-26.

98. Lauer GM, Walker BD. Hepatitis C virus infection. The New England journal of medicine 2001;345:41-52.

99. National Institutes of Health Consensus Development Conference Statement: Management of hepatitis C: 2002--June 10-12, 2002. Hepatology (Baltimore, Md 2002;36:S3-20.

100. Conry-Cantilena C, VanRaden M, Gibble J, et al. Routes of infection, viremia, and liver disease in blood donors found to have hepatitis C virus infection. The New England journal of medicine 1996;334:1691-6.

101. Kenny-Walsh E. Clinical outcomes after hepatitis C infection from contaminated anti-D immune globulin. Irish Hepatology Research Group. The New England journal of medicine 1999;340:1228-33.

102. Zignego AL, Craxi A. Extrahepatic manifestations of hepatitis C virus infection. Clin Liver Dis 2008;12:611-36, ix.

103. Hansson GK. Inflammation, atherosclerosis, and coronary artery disease. The New England journal of medicine 2005;352:1685-95.

104. Hansson GK, Libby P. The immune response in atherosclerosis: a double-edged sword. Nat Rev Immunol 2006;6:508-19.

105. Libby P. Inflammation in atherosclerosis. Nature 2002;420:868-74.

106. Butt AA, Fultz SL, Kwoh CK, Kelley D, Skanderson M, Justice AC. Risk of diabetes in HIV infected veterans pre- and post-HAART and the role of HCV coinfection. Hepatology (Baltimore, Md 2004;40:115-9.

107. Shaheen M, Echeverry D, Oblad MG, Montoya MI, Teklehaimanot S, Akhtar AJ. Hepatitis C, metabolic syndrome, and inflammatory markers: results from the Third National Health and Nutrition Examination Survey [NHANES III]. Diabetes Res Clin Pract 2007;75:320-6.

108. Koike K. Hepatitis C as a metabolic disease: Implication for the pathogenesis of NASH. Hepatol Res 2005;33:145-50.

109. Fukui M, Kitagawa Y, Nakamura N, Yoshikawa T. Hepatitis C virus and atherosclerosis in patients with type 2 diabetes. JAMA 2003;289:1245-6.

110. Volzke H, Schwahn C, Wolff B, et al. Hepatitis B and C virus infection and the risk of atherosclerosis in a general population. Atherosclerosis 2004;174:99-103.

111. Butt AA, Xiaoqiang W, Budoff M, Leaf D, Kuller LH, Justice AC. Hepatitis C virus infection and the risk of coronary disease. Clin Infect Dis 2009;49:225-32.

112. Lewis JD, Schinnar R, Bilker WB, Wang X, Strom BL. Validation studies of the health improvement network (THIN) database for pharmacoepidemiology research. Pharmacoepidemiol Drug Saf 2007;16:393-401.

113. Lis Y, Mann RD. The VAMP Research multi-purpose database in the U.K. J Clin Epidemiol 1995;48:431-43.

114. Chisholm J. The Read clinical classification. BMJ 1990;300:1092.

115. Garcia Rodriguez LA, Perez Gutthann S. Use of the UK General Practice Research Database for pharmacoepidemiology. Br J Clin Pharmacol 1998;45:419-25.

116. Lo Re V, 3rd, Haynes K, Forde KA, Localio AR, Schinnar R, Lewis JD. Validity of The Health Improvement Network (THIN) for epidemiologic studies of hepatitis C virus infection. Pharmacoepidemiol Drug Saf 2009;18:807-14.

117. Haynes K, Bilker WB, Tenhave TR, Strom BL, Lewis JD. Temporal and within practice variability in the health improvement network. Pharmacoepidemiology and drug safety 2011.

118. Hammad TA, McAdams MA, Feight A, Iyasu S, Dal Pan GJ. Determining the predictive value of Read/OXMIS codes to identify incident acute myocardial infarction in the General Practice Research Database. Pharmacoepidemiol Drug Saf 2008;17:1197-201.

119. Cohn JN, Levine TB, Olivari MT, et al. Plasma norepinephrine as a guide to prognosis in patients with chronic congestive heart failure. N Engl J Med 1984;311:819-23.

120. Cox D. Regression Models and Life Tables. Journal of the Royal Statistical Society Series 1972;B 34:187–220.

121. Rothman KJ, Greenland S. Modern Epidemiology. Second ed. Philadelphia: Lippincott Williams and Wilkins; 1998.

122. Rubin DB. Multiple imputation for nonresponse in surveys. New York: John Wiley; 1987.

123. Little RJA, Rubin DB. Statistical analysis with missing data. New York: Wiley; 1989.

124. StataCorp. Stata: Release 11. Multiple-imputation reference manual. College Station, TX: Stata Press; 2009.

125. Li KH. Imputation using Markov chains. Journal of Statistical Computation and Simulation 1988;30:57–79.

126. Lewis JD, Bilker WB, Weinstein RB, Strom BL. The relationship between time since registration and measured incidence rates in the General Practice Research Database. Pharmacoepidemiol Drug Saf 2005;14:443-51.

127. Solomon DH, Karlson EW, Rimm EB, et al. Cardiovascular morbidity and mortality in women diagnosed with rheumatoid arthritis. Circulation 2003;107:1303-7.

128. Gelfand JM, Neimann AL, Shin DB, Wang X, Margolis DJ, Troxel AB. Risk of myocardial infarction in patients with psoriasis. JAMA 2006;296:1735-41.

129. Nikpour M, Urowitz MB, Gladman DD. Premature atherosclerosis in systemic lupus erythematosus. Rheum Dis Clin North Am 2005;31:329-54, vii-viii.

130. del Rincon ID, Williams K, Stern MP, Freeman GL, Escalante A. High incidence of cardiovascular events in a rheumatoid arthritis cohort not explained by traditional cardiac risk factors. Arthritis Rheum 2001;44:2737-45.

131. Esdaile JM, Abrahamowicz M, Grodzicky T, et al. Traditional Framingham risk factors fail to fully account for accelerated atherosclerosis in systemic lupus erythematosus. Arthritis Rheum 2001;44:2331-7.

132. Ward MM. Premature morbidity from cardiovascular and cerebrovascular diseases in women with systemic lupus erythematosus. Arthritis Rheum 1999;42:338-46.

133. Marzouk D, Sass J, Bakr I, et al. Metabolic and cardiovascular risk profiles and hepatitis C virus infection in rural Egypt. Gut 2007;56:1105-10.

134. Eknoyan G. Adolphe Quetelet (1796-1874)--the average man and indices of obesity. Nephrol Dial Transplant 2008;23:47-51.

135. Jahoda G. Quetelet and the emergence of the behavioral sciences. Springerplus 2015;4:473.

136. Greenland S, Finkle WD. A critical look at methods for handling missing covariates in epidemiologic regression analyses. Am J Epidemiol 1995;142:1255-64.

137. Rubin DB. Inference and missing data. Biometrika 1976;63:581-92.

138. van Buuren S, Boshuizen HC, Knook DL. Multiple imputation of missing blood pressure covariates in survival analysis. Stat Med 1999;18:681-94.

139. Schneeweiss S, Rassen JA, Glynn RJ, Avorn J, Mogun H, Brookhart MA. High-dimensional propensity score adjustment in studies of treatment effects using health care claims data. Epidemiology 2009;20:512-22.

140. Witten IH, Frank E, Hall MA. Data Mining: Practical Machine Learning Tools and Techniques. Third ed: Elsevier; 2011.

141. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2008. at http://www.R-project.org.)