4-2015

Examining the Transitional Impact of ICD-10 on Healthcare Fraud Detection

Tyler Olson
College of Saint Benedict/Saint John's University

Examining the Transitional Impact of ICD-10 on Healthcare Fraud Detection

An Honors Thesis

College of St. Benedict / St. John's University

In Partial Fulfillment of the Requirements for Distinction in the Department of Computer Science

Tyler Olson

Advisor: Dr. Yu Zhang

April 30th, 2015

Approval Page

Approved By:

Dr. Yu Zhang, Advisor
Associate Professor of Computer Science

Dr. Phil Byrne, Reader
Professor of Mathematics

Mr. John Miller, Reader
Computer Science Department Lab Coordinator

Dr. Imad Rahal, Department Chair
Chair, Department of Computer Science

Dr. Emily Esch,
Director, Honors Thesis Program

Abstract

On October 1st, 2015, the tenth revision of the International Classification of Diseases (ICD-10) will be mandatorily implemented in the United States. Although this medical classification system will allow healthcare professionals to code with greater accuracy, specificity, and detail, these codes will have a significant impact on the flavor of healthcare insurance claims. While the overall benefit of ICD-10 throughout the healthcare industry is unquestionable, some experts believe healthcare fraud detection and prevention could experience an initial drop in performance due to the implementation of ICD-10. We aim to quantitatively test the validity of this concern regarding an adverse transitional impact. This project explores how predictive fraud detection systems developed using ICD-9 claims data will initially react to the introduction of ICD-10. We have developed a basic fraud detection system incorporating both unsupervised and supervised learning methods in order to examine the potential fraudulence of both ICD-9 and ICD-10 claims in a predictive environment. Using this system, we are able to analyze the ability and performance of statistical methods trained using ICD-9 data to properly identify fraudulent ICD-10 claims. This research makes contributions to the domains of medical coding, healthcare informatics, and fraud detection.

Table of Contents

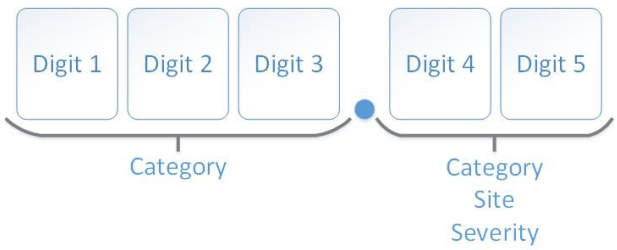
1 Introduction

1.1 Motivation

The process of medical classification and coding has been used for centuries to efficiently gather statistical data in an effort to measure the frequent causations of morality. The International List of Causes of Death, the first edition of international medical classification, was formally adopted in 1893. This classification system eventually evolved into the International Classification of Diseases (ICD), which has been maintained by the World Health Organization (WHO) since 1948. The ICD code set allows both mortality and morbidity conditions to be described and tracked, and ten revisions to this system have been published to date[1].

Today, when an individual is seen or treated by a healthcare professional, a series of alphanumeric codes are still used to describe the medical diagnoses and services provided. This designated classification structure, the ninth iteration of ICD, implements the use of coding for healthcare management, public health and medical informatics, and insurance purposes. ICD-9 has been the coding standard in the healthcare industry since October 1st, 1984. The primary purpose of ICD-9 is to translate written information from a patient's clinical statement regarding diagnoses and inpatient procedures into a series of universally understandable designations.

The ICD-9 code set contains approximately 13,000 distinct codes. Each ICD-9 code consists of a minimum of three digits and a maximum of five digits, with a decimal point after the third digit if more than three digits are used. Figure 1 illustrates the standard format of an ICD-9 code. The first three digits represent a single disease entity, or a group of similar or closely related conditions. The fourth digit identifies a subcategory, providing additional information regarding the etiology, site, or disease manifestation. Lastly, the fifth digit offers sub-classification of the subcategory, describing for example the mode of diagnosis or the anatomical site. ICD-9 is primarily numeric, with the exception of supplementary V-codes and E-codes. V-codes, characterized by a "V" as the first digit, are used when a patient seeks health care for reasons other than illness or injury. E-codes describe external causation of injury, poisoning, and adverse reactions, and where, why, and how an injury occurred. The structure of injuries described by ICD-9 codes are designated by the wound type, and the code omits laterality (left or right). An example of an ICD-9 code is 812.21, which describes a closed fracture of the shaft of the humerus. The first three digits, 812, describe a fracture of the humerus. The fourth and fifth digits offer greater detail, describing a closed fracture on the shaft.

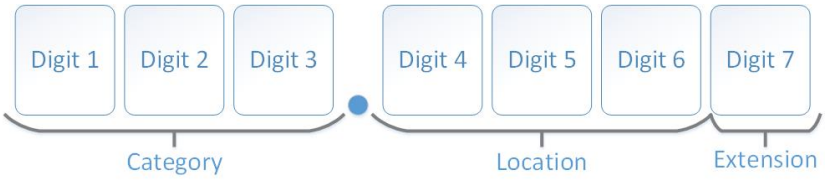


(1)

[1] http://www.who.int/classifications/icd/en/

On October 1st, 2015, the new ICD-10 diagnosis and procedure coding system will take effect in the United States. This implementation, which was originally scheduled to occur in October of 2011, has been delayed multiple times in an effort to allow the industry to become fully prepared. ICD-10 differs significantly from ICD-9 in terms of structure and organization. The composition of the updated clinical coding system promotes a higher level of detail and specificity, which will better capture and describe necessary medical data [36]. The additional information provided by ICD-10 should ideally allow all parties involved, i.e. patients, providers, clearinghouses, and insurance companies, to operate in a more effective and efficient manner.

The ICD-10 code set contains approximately 68,000 distinct codes, more than five times the number of possible ICD-9 codes. Each ICD-10 code consists of a minimum of three digits and a maximum of seven digits, with a decimal point after the third digit if more than three digits are used. Figure 2 illustrates the standard format of an ICD-10 code. The first three digits represent the category, the fourth, fifth, and sixth digits represent the location, and the seventh digit identifies an extension. Similar to ICD-9, the three digits to the immediate right of the decimal describe the etiology, anatomic site, and the severity. The seventh extension digit describes the visit encounter or sequel for injures and external causes. The first digit is always alphabetic, with the exception of the letter "U," the second digit is always numeric, and the remaining five digits are alphanumeric. The character "X" is used as a placeholder character, allowing for the future expansion of particular codes. V-codes and E-codes have been eliminated from the ICD-10 code set, and are now incorporated in the main code set. The structure of injuries described by ICD-10 codes are designated by the location/body part, and laterality is included. The ICD-10 equivalent of the ICD-9 code 812.21 described earlier is S42.321A. "S42" identifies the injury as a displaced transverse fracture. The number "3" specifies the fracture as a humerus fracture, the number "2" indicates the fracture is located on the shaft of the humerus, and the number "1" indicates the injury was sustained on the patient's right arm. An "A" extension signals that this is an initial encounter for the closed fracture for this particular patient.



(2)

Thirty years ago, when ICD-9 was first introduced, data needs were dramatically diminished. The applications for coded medical data today go well beyond the purposes for which ICD-9-CM was originally designed. The Centers for Medicare and Medicaid Services (CMS) have outlined the nine primary advantages ICD-10 will provide[2]. ICD-10 is superior to ICD-9 with respect to:

1. Measuring the quality, safety, and efficacy of care
2. Designing payment systems and processing claims for reimbursement
3. Conducting research, epidemiological studies, and clinical trials

[2] http://questions.cms.hhs.gov/app/answers/detail/a_id/10027/kw/icd-10

4. Setting health policy
5. Operational and strategic planning and designing healthcare delivery systems
6. Monitoring resource utilization
7. Improving clinical, financial, and administrative performance
8. Preventing and detecting healthcare fraud and abuse
9. Tracking public concerns and assessing risks of adverse public health events

The eighth item in the list above, the prevention and detection of healthcare fraud and abuse, is an aspect of ICD-10 that is studied throughout the course of this research project. The topic of healthcare fraud and abuse is discussed in further detail in the following subsection.

1.2 Background

Due to the nature of these coding languages, both ICD-9 and ICD-10 lend themselves to exploitation by physicians and/or providers. Social insurance programs such as Medicare and Medicaid, and private insurance companies allocate payment according to the clinical codes provided in healthcare claims. When codes are intentionally misrepresented, inappropriate monetary returns can potentially be distributed. Situations involving financially motivated deception are considered to be instances of healthcare fraud and abuse.

Fraud and abuse within the healthcare arena have occurred in numerous schematic forms, but there are nine primary strategies that have been identified as both prevalent and advantageous in the medical field [37]. These popular schemes are listed below.

1. Billing for services not rendered
2. Upcoding of services
3. Upcoding of items
4. Duplicate claims
5. Unbundling
6. Excessive services
7. Unnecessary services
8. Kickbacks and bribery

During this research project, we focused on the detection of billing for services not rendered, unbundling, and billing for excessive or unnecessary services. These particular schemes, numbers 1, 5, and 6, were chosen due to their potential affiliation with the medical claims data we had access to. The process of determining which types of fraud to concentrate on, referred to as goal setting, is a topic reviewed in Section 2.1.2.

In order to avoid the improper payment of fraudulent healthcare claims, the Centers for Medicare and Medicaid Services (CMS) and private companies such as Blue Cross and Blue Shield, United Healthcare, and Humana use aggregations of analytical algorithms for detection and prevention. These machine learning algorithms can generally be divided into two main categories: supervised learning and unsupervised learning. Both groups are discussed in further detail in Sections 2.2 and 2.3. Supervised statistical methods are commonly used to analyze current and historical data and patterns in an effort to accurately predict future, unknown events

and outcomes. The dynamic nature of these techniques allows a continuous increase in performance and accuracy to occur. When ICD-10 is introduced in October of 2015, these systems, constructed using past ICD-9 data and trends, will essentially be reset, and the accuracy of fraud detection could potentially deteriorate until a proper amount of ICD-10 data is accumulated [29]. The unknown impact of implementation is a serious threat to the predictive modeling necessary to properly identify fraudulent behavior [4, 18]. ICD-10 will undoubtedly benefit the entire industry once the transitional period has past, but healthcare insurance providers will encounter challenges during the infantile period regarding the accurate detection of fraud [8].

1.3 Research Goals

To the best of our knowledge, the future impact of ICD-10 on predictive fraud detection has yet to be studied in any quantifiable manner. Domain experts have suggested that the initial lack of ICD-10 training data could lead to the inaccurate recognition of abnormal medical claims, but no original, published analysis has examined this possibility. ICD-9 diagnosis and procedure codes have played an important role in the majority of past healthcare fraud research projects. Researchers have identified these codes as prevalent metrics that should be incorporated in the algorithms, models, and systems used to detect fraud [2]. However, the transition from ICD-9 to ICD-10 and the potential effect it will have these metrics has been generally overlooked. The objective of this research is to bridge the gap between existing healthcare fraud detection research and the industry's transition to ICD-10.

We evaluated this transitional impact through the use of logistic regression analysis, coupled with an outlier detection model. Due to the nature of healthcare claims data that is available to the public, data containing labels indicating fraud is extremely difficult to acquire. Claims identified as fraudulent by insurance companies and agencies are typically redacted from data that is eventually published, because they are both unlawful and illegitimate. Through the application of outlier detection, we were able to label medical claims data in an unsupervised manner for the purpose of supervised learning. The first phase of the experiment provides the labeled training data necessary to perform regression analysis. The development of a logistic regression model, the second phase, allowed us to study the influence of ICD-10 in a predictive environment.

This project explores the validity of predictions from domain professionals regarding fraud detection and the implementation of the ICD-10 code set. The notion that fraud detection systems using supervised learning algorithms will encounter an initial decline in performance due to ICD-10 is fairly unsupported at the moment. We claim that the results from our experiment will provide evidence that will support this notion of an initial negative transitional impact.

1.4 Approach

A year's worth of medical claims were first grouped into a smaller number of clinically meaningful categories according to their diagnosis and procedure codes. An outlier detection algorithm was then used to identify anomalies present in each of the groups based on the amount billed to a given payer, and those abnormal claims were flagged as fraudulent. The sole purpose of this process was to create the labeled training data necessary to construct a supervised learning model.

After the process of generating training data was complete, a logistic regression model served as a predictive tool that mimicked the predictive capabilities of existing fraud detection and prevention systems being used within the industry. A second year's worth of claims data was analyzed, and probable instances of fraud were flagged. ICD-9 codes, attributes of each claim, were translated into ICD-10 using crosswalks, and the modified dataset was re-analyzed. We found that eliminating two covariates caused the logistic regression models to flag a significant number of both false-positive and false-negative healthcare claims as fraudulent.

2 Related Work

2.1 Healthcare Data

The implementation of the electronic health record (EHR) has allowed healthcare organizations to collect and externally report/provide a greater amount of data to the public sector. The EHR is a systematic collection of electronic health information about an individual patient or population, and can include a range of data, including demographics, medical history, medication and allergies, immunization status, laboratory test results, radiology images, vital signs, personal statistics, and billing information. Fraud detection research in the field of healthcare management has primarily concentrated on medical billing data, which is derived from the EHR. After a patient's medical record is updated by a physician or staff member, diagnosis and procedure codes are assigned by a medical coder. The appropriate medical codes and necessary data from the EHR are incorporated into an ANSI 837 file, which is submitted to the payer directly or via a clearinghouse. An insurance company is usually the recipient of this claim file, so the majority of suitable fraud detection data comes from health insurance agencies.

2.1.1 Sources

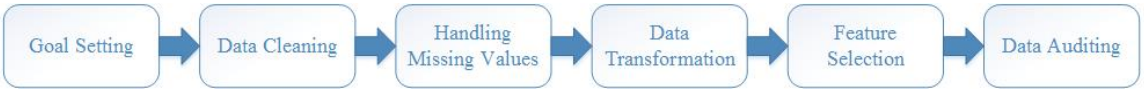
Healthcare fraud literature originating from countries outside of the United States have used a variety of sources to acquire medical claims data. The National Health Insurance Administration (NHIA) in Taiwan has provided data to multiple research groups [6, 16, 22, 38, 42], and the NHIA equivalent in South Korea, the National Health Insurance (NHI) system, has also contributed to studies [34]. Two major Australian governmental health departments, the Health Insurance Commission (HIC) and Medicare Australia, have been reported as the sources of data in numerous pertinent research projects [14, 15, 31, 32, 36, 39]. Healthcare claims data from private insurance companies located in Turkey and Chile has also been used by several researchers [17, 19, 28].

Within the United States, various agencies within the Department of Health and Human Services (HHS) have been involved with research exploring the detection and prevention of healthcare fraud. The Centers for Medicare and Medicaid Services (CMS) and its predecessor, the Health Care Financing Administration (HCFA), both supplied members of academia with Medicare and Medicaid data [11, 23, 25, 33]. Researchers in the United States have also worked with private insurance companies and hospitals, using data collected by these organizations [24, 27].

2.1.2 Preprocessing

The raw healthcare insurance data provided by any governmental health department or private agency is rarely organized in such a way that the researcher is satisfied with the structure. In order to appropriately arrange and organize the data, preprocessing must occur. The raw data must be processed into a form that is suitable for the statistical methods being used. Although this task is both extremely time-consuming and challenging, the process is infrequently documented, explicitly described in only two pieces of literature [21, 35]. Using aggregated information from these two papers and various other unrelated external sources, Li, Huang, Jin,

and Shi constructed the following flowchart, Figure 3, which outlines the steps commonly involved with preprocessing [20].



(3)

The purpose of goal setting is to determine which forms of healthcare fraud, identified in Section 1.2, are of interest and/or concern. Detection can then be tailored to focus on these particular schemes. Typically, domain experts assist in this process of prioritization, having both the knowledge and the resources to gauge the frequency and financial loss associated with different types of fraud. The medical claims data used during the experimentation phase must then correspond to the forms of fraud selected. This step is rarely discussed in relevant literature, but has likely occurred in some capacity during every documented research project involving healthcare fraud detection. Sokol, Garcia, West, Rodriguez and Johnson met with representatives from the HCFA and the Office of the Inspector General, and ultimately decided to focus on six fraudulent schemes that could be properly identified using available HCFA data [35]. Capelleveen used the input provided by Medicaid fraud experts to zero-in on the most prevalent schemes found in Medicaid dental claims [5].

Flaws within healthcare claims data and irrelevant views of the data can cause significant issues during the training of a statistical model, the grouping of abnormal claims, or the identification of suspicious instances. Resolving inconsistent representations of the same concept, appropriately handling missing values, and transforming raw data into a flattened table format are all vital steps of the preprocessing phase. The data used in our research had already undergone cleaning, imputation, and transformation by the vendor. Therefore, these three stages of the preprocessing method will not be discussed.

In order to maximize the discrimination power of any given statistical method being employed in a fraud detection capacity, features, also known as metrics or predictors, must be selected from the original data attributes that will provide the information necessary to separate fraudulent and legitimate claims. Feature selection can be done manually with the assistance of domain experts, or computationally using machine learning algorithms. Most researchers working in the healthcare domain consult with experts to identify discriminating metrics. Due to the sensitive nature of this classified information, this process is usually redacted in the literature by the authors. Ortega, Figueroa, and Ruz published the procedure they used, but were unable to include the features that they ultimately selected [28]. This group worked with domain experts to first define a preliminary set of features, correlation checks were then performed to delete redundant features, and lastly, the discriminating power of each feature was tested. Only those features with discriminating power above a certain predefined threshold were selected.

The auditing of data to assess both quality and utility is an important conclusion to the preprocessing method. Researchers use statistical software such as SAS, SPSS, Stata, and R to execute basic statistical analysis and visualization in an effort to become familiar with the data. Every research group referenced so far performed some form of data auditing before conducting their respective experiments.

2.2 Statistical Modeling Involving Supervised Learning

Supervised learning is a category of machine learning algorithms that uses training data to make predictions. Training data consists of both input data as well as corresponding response values. Using this known information, supervised learning algorithms develop models that can predict response values for instances in an unknown set of data. These algorithms benefit from larger sets of training data, which allows them to construct models with higher predictive power that can generalize more accurately.

2.2.1 Support Vector Machines

Using a set of training data where each instance is identified as being fraudulent or non-fraudulent, a Support Vector Machine (SVM) constructs a model that assigns new healthcare claims to one of the two categories. The SVM creates a hyperplane, and each training data instance is represented as a point in space. The functional margin, the distance to the nearest training instance in either category, should be as large as possible. When the functional margin is maximized, a clear gap will exist between the two categories, which will minimize the generalization error of the SVM. Each new claim is mapped onto the hyperplane, and its location in the functional margin determines if the claim is deemed fraudulent or non-fraudulent. Both research groups that have used SVMs relied on the standard linear classification ability of the SVM to detect abnormal healthcare claims. Kirlidog and Asuk used longitudinal data that spanned a nine year period, marking records that had a probability of anomaly greater than 0.5 as anomalous [17]. Of the 808,348 records spanning from 2001 to 2009, 6,595 claims had probabilities ranging from 0.5 to 0.673. These anomalous claims were analyzed according to three primary criteria: the status of the claim (rejected or accepted), the excessivity in terms of the bill amount of the claim compared to other claims from the same type of health center, and the excessivity of the claim compared to other claims from a particular health center. The purpose of this analysis was to examine the rejected-anomalous relationship and the possibility of initiating investigations based off the relative excessivity of claims. Kumar, Ghani, and Mei addressed concept drift, the scenario when the relation between the input data and the target variable changes over time, and the evolution of the target function by using seasonal subsets of their data, instead of longitudinal data [19]. A system was proposed that minimized payment errors made by insurance companies by predicting claims that needed to be reworked using SVMs. They found that this system produced an order of magnitude better precision over existing detection approaches, and this in turn could potentially save insurance companies $15 to $25 million each year.

2.2.2 Artificial Neural Networks

An Artificial Neural Network (ANN) is a graph consisting of nodes and edges that is organized into layers. Each layer is made up of a number of interconnected nodes which contain an activation function. Training data containing known instances of fraud is presented to the ANN via the input layer, which in turn communicates to one or more of the hidden layers. The adaptive weights of the edges are tuned by a learning algorithm according to the input training data. The hidden layers are connected to the output layer, which identifies new records as fraudulent or non-fraudulent. ANN's are generally used to approximate functions that are

unknown and depend on a large number of inputs. In the case of fraud detection, the normality of each new claim is unknown, and the input is typically a large collection of medical claim features. A two-layer neural network, the standard Multi-Layer Perceptron (MLP) consisting of an input layer, a hidden layer, and an output layer, is the most-commonly used ANN. Oretga, Figueroa, and Ruz originally modeled each fraud problem using a standard MLP and small hidden layers, but the variance of each model was too high [28]. A committee of 10 multi-layer neural networks replaced the standard MLP, and the variance was appropriately reduced. Their proposed system assigned a committee to each of the four entities primarily involved with healthcare fraud: providers, medical employers, affiliates, and medical claims. The implementation of ANNs allowed all four types of fraudulent entities to be identified at a significantly quicker rate, thereby reducing the loss of insurance revenue. During a comparison of the healthcare fraud detection performance of neural networks, logistic regression models, and classification trees, the two-layer neural network was tested against the two alternative supervised learning methods [22]. Using Clementine neural networks, variables were ranked according to their classification importance through sensitivity analysis, and these rankings were used in the construction of the ANN. The neural network algorithm correctly identified 100% of the fraudulent hospitals, 91.47% of the normal hospitals, and had an overall correct identification rate of 95.73%.

2.2.3 Classification Trees

Decision trees, which model decisions and their possible consequences, have been used as a predictive tool to map features of healthcare claims to conclusions regarding fraudulence and abnormality. When the target variable, indicating fraud or non-fraud, can only assume a finite set of values, the decision tree is considered to be a classification tree. Within the structure of a classification tree, external, leaf nodes represent the class labels of fraud and non-fraud, while branches represent conjunctions of features from a medical claim that lead to those two class labels. Each internal, non-leaf node is labeled with an input feature from a claim instance, and the directed edges emanating from these nodes are labeled with each of the possible values the features can potentially assume. Training data is used to construct an appropriate classification tree, and the fraudulent nature of unidentified healthcare claims is predicted using this logic model. Classification trees have been used by researchers in a comparative capacity to identify fraudulent reporting of diabetic outpatient services, tested as an auditing strategy in the fiscal and insurance domains, and generated using the C4.5 and C5.0 classification algorithms to detect insurance subscribers' fraud. Liou, Tang, and Chen found that a classification tree correctly identified 100% of the fraudulent hospitals present in a set of data, 98.73% of the normal hospitals in the dataset, and had an overall correct identification rate of 99.3% [22]. A methodology for constructing profiles of fraudulent paying entities was proposed by Bonchi, Giannotti, Mainetto, and Pedreschi [3]. The following methodological issues were identified in this paper: defining of an audit cost model, monitoring the training-set construction, measuring the quality of a classifier, and tuning the classifier construction. By properly addressing these issues, the researchers were able to develop an effective decision support system for audit planning. The hot spots methodology, introduced by Williams and Huang, involved clustering to provide a first cut segmentation of the data [40]. Using C4.5 and C5.0, decision tree induction and rule set pruning then assigned a rule to each segment of the data. These rules, coupled with

the original data, were analyzed to find "nuggets," subsets of the original data, which were related in some way to the domain problem.

2.2.4 Logistic Regression

Shin, Park, Lee, and Jhee proposed a scoring model for a South Korean governmental health insurance agency that detected outpatient clinics with abusive utilization patterns based on profiling information extracted from electronic insurance claims [34]. Their model consisted of scoring claims to quantify the degree of abusiveness and segmentation to categorize the problematic providers with similar utilization patterns. Practitioner claims submitted to the South Korean National Health Insurance Corporation (NHIC) for outpatient care during the 3rd quarter of 2007 were used to construct the model, and data from the 4th quarter of 2007 was used to validate the model. They compared the conditional probability distributions of the composite degree of anomaly (CDA) score formulated for intervention and non-intervention groups. The CDA aggregated 38 indicators of abusiveness for individual clinics, which were grouped based on the CDAs. This combination of logistic regression and CDAs allowed Shin, Park, Lee, and Jhee to improve upon the performance of existing fraud detection methods. As previously mentioned, the detection performance of logistic regression was analyzed alongside neural networks and classification trees by Liou, Tang, and Chen [22]. They determined while classifications trees had an overall correct identification rate of 99%, neural networks had an overall correct identification rate of 96%, and logistic regression had an overall correct identification rate of 92%, all three algorithms performed quite accurately.

We chose to implement logistic regression models as the supervised, predictive element of our fraud detection system due to their comparative simplicity. Although both classification trees and neural networks were found to be more accurate, we discovered that they were much more difficult to develop and implement. The straightforwardness of this statistical model allowed us to closely monitor the addition and subtraction of covariates, and there were multiple existing functions in R packages that performed logistic regression.

2.3 Statistical Modeling Involving Unsupervised Learning

Unsupervised learning is a category of machine learning algorithms that uses unknown data to draw inferences. An unknown dataset consists of input data instances without labeled responses, meaning the latent variable is unknown. Using only the information provided by the input attributes, unsupervised learning algorithms develop models that attempt to discover hidden structure in the unlabeled data. Values are assigned to the response variable according to the structure and patterns of the data that are found by the unsupervised learning algorithm.

2.3.1 Cluster Analysis

The purpose of cluster analysis is to group data into categories, classes, or clusters, so that items within a particular cluster are similar in comparison to one another, but significantly dissimilar to items in other clusters. By clustering elements of a set into two or more mutually exclusive groups, it becomes more manageable to predict behavior or properties based on group membership. A data matrix must first be constructed, where each medical claim is represented by a row, and each feature of the claim is stored a column. Then, a table of relative similarities or differences between all the claims can be developed, which is called the proximities matrix. Within this matrix, both the rows and columns represent individual claims, and the value of each element is a measurement of the similarity or difference between the two particular claims. The measure of similarity on which the clusters are eventually modeled can be defined by Euclidean distance, probabilistic distance, or other appropriate metrics. After the distances between all the claims have been found, clustering occurs based on these distances. A clustering algorithm, such as Hierarchical clustering (HCA), k-means clustering, Gaussian mixture models, or Self-organizing maps (SOM), is responsible for properly dividing the claims into clusters. In the application of fraud detection, claims within clusters containing seemingly abnormal feature values are typically flagged as fraudulent. Researchers have used various clustering algorithms to detect different forms of healthcare fraud through cluster analysis. Demographically homogenous zip code regions were created using clustering procedures, and each zip code region was associated with a random variable that could discriminate between health care utilization. [25]. He, Graco, and Yao coupled the K-nearest neighbor (KNN) algorithm with a genetic algorithm to detect Australian medical fraud [14]. The genetic algorithm determined the optimal weights of features used to categorize General Practitioners' practice profiles, and the KNN algorithm used these weights to identify nearest neighbors. The results of this experiment were promising, and the researchers recommended the implementation of this model in the Health Insurance Commission's fraud prevention system (fps). Within the UNISIM system, proposed by Tang, Mendis, Murray, Hu, and Sutinen, the framework consisted of a feature extractor, a cluster builder, a model constructor, and an outlier detector [36]. The purpose of the cluster analysis within this system was to examine and label the data according to certain criteria. Every sequence was first initialized as a cluster, nearest neighbors were merged based on density of sequences, and a second merging process occurred based on the density of the clusters. Overall, the unsupervised UNISIM system was proven to be an effective, yet complementary tool in the detection of healthcare fraud.

2.3.2 Association Rules

In an elemental sense, association rules are if/then statements that allow relationships between seemingly unrelated data attributes to be exposed. An antecedent, the "if", and a consequent, the "then", are the two components of an association rule. Antecedents are the values of certain data attributes, and consequences are other feature values that are found in combination with a particular antecedent. The association rules themselves are developed by analyzing the data for frequent reasoning patterns, and then using the criteria support and confidence to determine the most significant relationships. Criteria support indicates how frequently the antecedent and consequent appear in the data, and confidence indicates the number of times the antecedent/consequent combination has been found to be true. In the application of fraud detection, medical claims that lack correspondence to any existing association rules would be flagged as abnormal and further investigation would occur. The detection of provider fraud through specialist billing was studied by Shan, Jeacocke, Murray, and Sutinen using positive and negative association rules [31]. Rules were first identified by the researchers, and then classified into two groups representing compliance and non-compliance by a domain expert. Any claims that were not consistent with the compliance rules were considered to be potentially fraudulent. This method was tested against a baseline classifier, and these rules were validated after significantly outperforming the baseline. Another Australian researcher, Williams, coupled rule induction and clustering to detect various forms of insurance subscribers' fraud [39]. This research is closely related to research that Williams conducted with Huang two years earlier in 1997. The results from [40] have already been discussed in Section 2.2.3.

2.3.3 Anomaly Detection

Anomaly detection, or outlier detection, is the identification of cases which do not conform to an expected pattern, or that are unusual within data that is seemingly homogeneous. This statistical method performs quite well as a detection tool, because it was developed to recognize rare events that may have great significance, but are hard to find within a large set of data. Unsupervised anomaly detection techniques detect anomalies in an unlabeled test data set under the assumption that the majority of the instances in the data set are normal by looking for instances that seem to fit least to the remainder of the data set. The distance from the center of a normal distribution indicates how typical a given point is with respect to the distribution of the data. Each case can be ranked according to the probability that it is either typical or atypical. Capelleveen tested the performance of an outlier detection method by analyzing its precision predicting dental fraud [5]. Local density based outlier detection was used by Shan, Murray, and Sutinen to study fraudulent Optometrist billing patterns, and by Tang, Mendis, Murray, Hu, and Sutinen in conjunction with feature selection, clustering, and pattern recognition in their proposed UNISIM system [32, 36].

Ngufor and Wojtusiak established the unsupervised data labeling capability of outlier detection, and proposed a SynTwoMoving algorithm to label fraudulent healthcare data, which incorporated concept drift [27]. Using this research as inspiration, we decided to label our training data using an anomaly detection algorithm that incorporated this aspect of seasonality.

3 Data Source and Preprocessing

3.1 Methodology

Since the emergence of the Electronic Health Record (EHR) and the subsequent increase in the electronic submission of healthcare insurance claims, several states have established databases that collect health insurance claims information from all paying entities into a statewide information repository. The All-Payer Claims Databases (APCD) contain medical, dental, and pharmaceutical claims data that can be used to report cost, use, and quality information. The stored data is service-level information based on claims that have been processed by various payers. Information considered to be service-level includes charges and payments, the providers receiving payment, clinical diagnosis and procedure codes, and patient demographics. The various payers include private health insurance companies, federally-funded agencies such as Medicare and Medicaid, state employee health benefit programs, prescription drug plans, dental insurance companies, and self-insured employer plans. At this time, ten states have existing All-Payer Claims Databases: Colorado, Kansas, Maine, Massachusetts, Maryland, Minnesota, New Hampshire, Tennessee, Utah, and Vermont. Most states offer limited use datasets containing certain identifying features, and public use datasets that ensure patient privacy by encrypting, aggregating, or suppressing all patient identifiers. We obtained the public use APCD dataset for this research project from the New Hampshire Comprehensive Health Care Information System (NHCHIS).

Once we received the dataset from Milliman, the vendor responsible for collecting, cleaning, imputing, and transforming the data from the New Hampshire APCD, our first course of action was to determine the types of fraud we could potentially detect using the information provided by attributes in the data. After performing some basic analysis to become familiar with the data, we decided to focus on detecting claims that billed for services not rendered or billed for excessive or unnecessary services. The results from this initial data analysis are discussed in Section 3.2.

Features of the dataset were then selected that would provide the algorithms being used with the necessary information to identify instances of these fraudulent schemes. The predicting capability of each available attribute was evaluated using literature written by domain experts and the methodologies from previous research projects. The results of the feature selection process, which provided the data elements for both the outlier detection and logistic regression models, are included in Sections 5 and 6.

During the data auditing process, we learned that the vendor had redacted the exact submission date of each claim due to privacy concerns. Time-series data was an input requirement for the outlier detection package that had been chosen, so this lack of any temporal indicator was troubling. Thankfully, we were able to determine through additional analysis that the imputed service key value, generated by the vendor, was assigned to each claim according to when the claim was processed and stored. This value was used to create a sort of pseudo-time-stamp, and allowed us to organize the data in a sequential manner.

The amount billed for each service record was also evaluated, and we discovered that this currency field can be negative when a claim is reversed. Healthcare fraud is monetarily motivated, so these type of records were eliminated from the dataset.

3.2 Data Analysis

The following experiment was conducted using a dataset from the New Hampshire Comprehensive Health Care Information System (NHCHIS), an APCD system that is maintained by the New Hampshire Insurance Department and the New Hampshire Department of Health and Human Services. The NHCHIS began accepting claims submissions from paying entities in 2005, and currently collects medical claims data from commercial payers, third-party administrators, Medicaid, and Medicare. We originally requested and received public use data from 2005 through 2014, but eventually focused on claims from 2012 and 2013, due to their recentness and completeness.

After importing the two years' worth of NHCHIS information into a SQL Server database, the structure of the data was analyzed. Each row in the table represents a service that was provided and is being billed for by a provider within a healthcare insurance claim. An individual service record in a claim can be uniquely identified by a service key, which was generated by the warehouse during the data transformation process. Medical claims, containing one or more service records, can be uniquely identified by a claim key that was assigned by Milliman. Each claim describes the services rendered by a healthcare provider for an individual patient during a particular period of time.

Including the claim and service keys, a single record is comprised of 63 data attributes that describe the patient, the provider, the situation, the services provided, billing information, insurance information, etc. Seven of the elements are identification keys that were generated by the vendor for confidentiality or reference purposes. The other 56 elements of each row are directly extracted from the claims data supplied by the healthcare insurance agencies and companies cooperating with the NHCHIS.

Outpatient claims describing care that was provided to a patient who was not formally admitted to a healthcare facility can be distinguished from inpatient claims according to certain data attributes. Records containing non-null ICD-9 Procedure codes (ICD_PROC_01_PRI) are classified as inpatient, while records with non-null CPT/HCPCS codes (PROC_CODE) are considered to be outpatient. An inpatient flag (INPATIENT_FLAG) also indicates whether a given service record is from an inpatient claim.

This project is motivated by the transition from ICD-9 to ICD-10, and inpatient claims face a far more dramatic change, since both the diagnosis and procedure codes will be transitioning. Therefore, we excluded outpatient claims from this experiment, and concentrated solely on testing inpatient claims.

4 Clinical Classification Software

4.1 Clinical Classification Methodology

Developed by the Agency for Healthcare Research and Quality (AHRQ), Clinical Classifications Software (CCS) is a tool for clustering patient diagnoses and procedures into a more manageable number of clinically meaningful categories. CCS provides a way to classify diagnoses and procedures into a limited number of categories by aggregating individual ICD-9 and ICD-10 diagnosis, ICD-9 and ICD-10 procedure, and Current Procedural Terminology (CPT)/Healthcare Common Procedure Coding System (HCPCS) codes into broad diagnosis and procedure groups to facilitate statistical analysis and reporting. This grouping process makes it easier to understand patterns of diagnoses and procedures so that organizations and researchers can analyze costs, utilization, and outcomes associated with particular illnesses and procedures. Single-level CCS categories, which are mutually exclusive, can be employed in many types of projects analyzing data on diagnoses and procedures. For example, they can be used to[3]:

- Identify cases for disease-specific or procedure-specific studies
- Gain a better understanding of an institution's or health plan's distribution of patients across disease or procedure groupings
- Provide statistical information on characteristics, such as charges and length of stay, about relatively specific conditions
- Cross-classify procedures by diagnoses to provide insight into the variety of procedures performed for particular diagnoses.

4.1.1 ICD-9 and ICD-10 Diagnosis Codes

The single-level ICD-9 and ICD-10 diagnosis classification schemes both aggregate mortality and morbidity into 285 mutually exclusive categories, most of which are clinically homogeneous. Some heterogeneous categories are necessary; these combine several less common individual conditions within a body system. Table 1 provides an example of a row in the crosswalk from a group of analogous ICD-9 diagnosis codes to a single-level CCS category.

Table 1: Example of the Single-Level CCS Diagnosis Crosswalk

Description	ICD-9 Diagnosis Codes	CCS Category
HIV infection	042 0420 0421 0422 0429 0430 0431 0432 0433 0439 0440 0449 07953 27910 27919 79571 7958 V08	5

[3] https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp

4.1.2 ICD-9 and ICD-10 Procedure Codes

Single-level ICD-9 and ICD-10 procedure classification schemes both contain 231 mutually exclusive categories. ICD-9 and ICD-10 procedure codes are only used to describe services in inpatient records and claims. The term inpatient refers to any procedure that requires a patient to be admitted to a hospital, so the patient can be closely monitored during the procedure and recovery process. Many of the categories represent single procedures; however, some procedures that occur infrequently are grouped according to three dimensions: the relevant body system, whether they are used for diagnostic or therapeutic purposes, and whether they are considered operating room or non-operating room procedures, according to diagnosis related groups (DRG) definitions. Table 2 provides an example of a single row from the ICD-9 CCS procedure crosswalk.

Table 2: Example of the Single-Level CCS ICD-9 Procedure Crosswalk

Description	ICD-9 Procedure Codes	CCS Category
Heart Valve Procedures	3500 3501 3502 3503 3504 3505 3506 3507 3508 3509 3510 3511 3512 3513 3514 3520 3521 3522 3523 3524 3525 3526 3527 3528 3596 3597 3599	43

4.1.3 CPT/HCPCS Codes

The CCS also provides a crosswalk for classifying Current Procedural Terminology (CPT) codes and Healthcare Common Procedure Coding System (HCPCS) codes into procedure categories. CPT, also referred to as HCPCS Level I, is used to describe outpatient procedures performed by healthcare professionals. An outpatient procedure does not require hospital admission, and may be performed off-site. HCPCS, also referred to as HCPCS Level II, is a supplementary coding system developed by the CMS to designate supplies and services not accounted for in the CPT code set. More than 9,000 CPT codes and 6,000 HCPCS codes are grouped into 244 categories. Of these 244 categories, 231 are identical to the ICD-9 procedure categories, and 13 are specific groups unique to the service and supply codes in the CPT/HCPCS coding system. Instead of providing a crosswalk for each individual CPT/HCPCS code, CCS-Services and Procedures classifies according to ranges of code values, which is illustrated in Table 3.

Table 3: Example of the CCS-Services and Procedures Crosswalk

CPT/HCPCS Codes	CCS Category	Description
'71250 – 71275' '75571 – 75573' 'S8032 – S8032' 'S8093 – S8093'	178	CT scan chest

In every CCS crosswalk, the diagnosis and procedure codes are represented with implicit decimals, which is fairly regular in the vast majority of healthcare data. In practice, ICD-9 and ICD-10 codes are usually represented with explicit decimals, as mentioned in Section 1.1.

4.2 Clinical Classification Results

For this experiment, the claims data from 2012 was used as training data, and the claims data from 2013 was used as the testing data. Using the CCS clinical classification schemes, each service record was categorized according to the primary diagnosis and primary procedure codes. This was done to appropriately group service records that have similar medical codes into distinct categories that would have similar bill amounts and quantities of services provided. Statistical analysis could then be performed in the future stages according to the CCS diagnosis and CCS procedure categories. The crosswalks described in Section 4.1 were used to assign each inpatient service record a diagnosis group number and a procedure group number. After all the inpatient claims from both years were grouped according to CCS Diagnosis/CCS Procedure combinations, the four groups with the largest number of claims, services, distinct diagnosis codes, and distinct procedure codes from 2012 were ultimately selected.

Table 4: 2012 IDC-9 Clinical Classification

CCS Diagnosis	CCS Procedure	Number of Claims	Number of Services	Number of Distinct ICD-9 Diagnosis Codes	Number of Distinct ICD-9 Procedure Codes
203	152	1569	22845	10	2
193	140	1027	7640	11	2
205	158	647	8157	22	13
149	84	585	8180	26	4

The same four CCS combinations from the 2013 inpatient claims were then selected, and the service records from each group were stored in a separate database table.

Table 5: 2013 ICD-9 Clinical Classification

CCS Diagnosis	CCS Procedure	Number of Claims	Number of Services	Number of Distinct ICD-9 Diagnosis Codes	Number of Distinct ICD-9 Procedure Codes
203	152	1959	28686	8	2
193	140	990	7172	8	3
205	158	607	7930	26	13
149	84	475	6540	25	3

5 Outlier Detection

5.1 Outlier Detection Methodology

In January 2015, Twitter released an open-source R package, *AnomalyDetection*, which has the ability to detect anomalies in big data. Considered to be both practical and robust, this package is intended to identify outliers in a set of time series data, and is cognizant of both seasonality and underlying trends. Positive, negative, global, and local anomalies can all be detected using the *AnomalyDetection* package. The use of time series decomposition and a robust statistical metric allows the package to detect this range of anomalies. The underlying algorithm, referred to as Seasonal Hybrid ESD, builds upon the Generalized ESD test for detecting anomalies. For long time series, the algorithm employs piecewise approximation, since the issue of trend extraction in the presence of anomalies is non-trivial, during the detection process.

This package was originally designed for time series data, but can also be used to detect anomalies in a vector of numerical values. This is extremely useful when the data is ordered according to time, but the corresponding timestamps are not available. *AnomalyDetection* allows the user to specify the direction of anomalies, the window of interest, enable/disable piecewise approximation, and annotate the axes to assist in visual data representation and analysis. The framework of Twitter's *AnomalyDetection* package is explained in the following subsections.

5.1.1 Student T-Distribution

Suppose we select a random sample of size n from a normal population with mean μ and variance σ^2. Let $\overline{x}$ represent the sample mean, and s represent the sample standard deviation. Then the random variable:

$$T_{n-1} = \frac{\overline{x} - \mu}{\frac{s}{\sqrt{n}}} \text{ has a t-distribution with } n - 1 \text{ degrees of freedom.} \quad (1)$$

This probability distribution is typically used to estimate population parameters when the population variance is unknown, or the sample size is small. When n is large ($n > 40$), a t-statistic will approximately follow a normal distribution, according to Central Limit Theory. Often, the population standard deviation is unknown, but the sample standard deviation can be calculated. The t-distribution allows researchers to conduct statistical analysis on this type of datasets using the normal distribution. As the degrees of freedom increases, the t-density approaches the normal density. This corresponds to the fact that the sample standard deviation s approaches the population standard deviation σ for large n. When the degrees of freedom is equal to 1,000, the critical values for the t-distribution are extremely close to the critical z-values (1.962 versus 1.96, for example).

T-density curves are also symmetric and bell-shaped like the normal distribution. However, the spread is more than that of the normal distribution. This is due to the fact that in equation 4, the denominator is *s* rather than σ. Since *s* is a random quantity varying with various samples, T_{n-1} has a higher degree of uncertainty, resulting in a larger spread.

5.1.2 Generalized ESD Test

The Generalized Extreme Studentized Deviate (ESD) test is used to detect one or more outliers in univariate data that follows an approximately normal distribution. The primary limitation of the Grubbs' test, the standard ESD test, is that the suspected number of outliers, *k*, must be specified exactly by the researcher. If *k* is not set correctly, this can cause the conclusion of the test to be misleading. The generalized derivation of the ESD test only requires that an upper bound for the suspected number of outliers be specified. Given this upper bound, *r*, the Generalized ESD test performs *r* separate tests: a test for one outlier, a test for two outliers, a test of three outliers, etc., testing up to *r* outliers.

The Generalized ESD test is defined using the following null and alternative hypotheses:

H_0: There are no outliers in the dataset
H_1: There are between 1 and *r* outliers in the dataset

Test Statistic:

$$R_i = \frac{max_i \, |x_i - \overline{x}|}{s} \qquad (2)$$

For the test statistic R_i, $\overline{x}$ and *s* denote the sample mean and the sample standard deviation. The numerator $max_i \, |x_i - \overline{x}|$ indicates the point farthest away from the sample mean.

Using the sample data, the observation that maximizes $|x_i - \overline{x}|$ is removed, and the test statistic is recomputed with *n* - 1 observations, resulting in the recalculation of the sample mean and sample standard deviation. This iterative procedure is repeated until *r* observations have been removed. At the conclusion of this process, *r* test statistics have been constructed, R_1, R_2, R_3 …, R_r.

Critical Region: For each of the *r* test statistics:

$$\lambda_i = \frac{(n-i)\, t_{p,n-i-1}}{\sqrt{\left(n-i-1+ t^2_{p,n-i-1}\right)(n-i+1)}} \qquad i = 1, 2, 3, \ldots, r \qquad (3)$$

The variable *n* represents the sample size of the dataset, and $t_{p,n-i-1}$ is the 100*p* critical value of the t-distribution with $n - i - 1$ degrees of freedom. The variable *p* is equal to:

$$p = 1 - \frac{\alpha}{2\,(n-i+1)} \qquad \alpha = \text{designated significance level} \qquad (4)$$

The number of outliers is ultimately determined by finding the largest i, such that the test statistic R_i is greater than the critical value λ_i.

5.1.3 Seasonality

The Generalized ESD test requires input data that follows an approximately normal distribution, but real-world time series data frequently has seasonality, regular or semi-regular cyclic variations that can affect statistical analysis. Seasonal Trend Decomposition using Locally-Weighted Scatterplot Smoothing (STL), designed to handle fixed seasonal periods, splits time series data into three separate elements: a seasonal component, a trend component, and a remainder component. The seasonal component of the original data is found using Locally-Weighted Scatterplot Smoothing (LOESS). The primary purpose of LOESS is the removal of "jaggedness" from data using local regression. These seasonal values are removed from the data, and the trend component is found by smoothing the remaining data. The remainder component, the residual data produced by removing both the seasonal and trend components, follows an approximately normal distribution. The Generalized ESD test can then be applied to this remaining data in order to detect anomalies.

5.2 Outlier Detection Results

The *AnomalyDetectionTs* function, found in Twitter's *AnomalyDetection* package, detects anomalies in seasonal univariate time series data where the input is a series of <timestamp, observation> pairs. This technique requires the timestamp to be in the YYYY-MM-DD HH-MM-SS format. Our pseudo-timestamp generated from the imputed service key did provide a temporal indictor, but was not in a standard timestamp format. Therefore, the *AnomalyDetectionVec* function, which does not require a timestamp input, was used instead. Outliers can still be identified in seasonal univariate time series data, but the input is a series of observations in sequential order.

According to previous methodologies, most supervised learning fraud detection systems that have been used by researchers rely on training data that has each claim, not each service record within a claim, identified as fraudulent or non-fraudulent [8, 9]. Therefore, during this data labeling process, the sum of the amounts billed divided by the quantities for the service records within an individual claim was identified as the primary feature for the univariate outlier detection process. Dividing the amount billed by the quantity of services, and summing these results for each claim provided a normalized total amount value that could be properly compared to other total values.

For each of the four CCS Diagnosis/Procedure combinations, the claims were ordered in ascending order according to their imputed service key. After the claims were in proper chronological order, the corresponding sums of the amounts billed divided by the quantities for the claims were stored in four separate R dataframes. The *AnomalyDetectionVec* function was then used to detect anomalies according to the total bill amount. Claims that were identified as anomalies by this package were marked as fraud in the SQL Server database.

Table 6: Anomaly Detection Variables		
Parameter	Description	Parameter Value
X	Time series as a column data frame, list, or vector, where the column consists of the observations	List of $\sum \frac{\text{Amount Billed}}{\text{Quantity}}$
max_anoms	Maximum number of anomalies that S-H-ESD will detect as a percentage of the data	0.05 (5%)
direction	Directionality of the anomalies to be detected	Positive
alpha	The level of statistical significance with which to accept or reject anomalies	0.05
period	Defines the number of observations in a single period, and used during seasonal decomposition	$\frac{\text{\# of Claims}}{12}$

The *AnomalyDetectionVec* function requires a number of input values that were subsequently used in the Seasonal Hybrid ESD test. The maximum number of anomalies as a percentage of the data was set to 0.05, or 5%, for this detection process. This threshold corresponds to the results published by Liou, Tang, and Chen, which indicated that 3% to 4% of healthcare claims were fraudulent or abusive [22]. Other research groups uncovered similar levels of fraud, ranging anywhere from 0.8% to 10% [17, 28]. We did not dynamically test this threshold value during our experiment, because this process of detecting outliers was used simply for unsupervised labeling for supervised learning purposes, not for the final detection of fraudulent instances. The directionality of the anomalies was set as positive, since healthcare fraud is motivated by monetary gain. The alpha level was set as 0.05, indicating that anomalies would be selected or rejected with a 95% confidence level. The period, which defines the number of observations per period and is used during seasonal decomposition, was determined by dividing the total number of claims by twelve. The anomaly results for each group can be found below in Table 7.

Table 7: CCS Outlier Detection Results							
CCS Diagnosis	CCS Procedure	Fraudulent Claims	Total Claims	Percent Fraudulent	Fraudulent Services	Total Services	Percent Fraudulent
203	152	13	1569	0.83	227	22845	0.99
193	140	51	1027	4.97	498	7640	6.52
205	158	29	647	4.48	309	8157	3.79
149	84	10	585	1.71	323	8180	3.95
TOTALS:		103	3828	2.69	1357	46822	2.90

6 ICD-9 and ICD-10 General Equivalence Mappings

6.1 General Equivalence Mapping Methodology

General Equivalence Mappings (GEMs) are medical coding crosswalks that were developed and published by the Centers for Medicare and Medicaid Services[4]. With the implementation date of the tenth medical classification revision nearing, GEMs are practical and useful translational dictionaries that provide acceptable ICD-10 alternatives to ICD-9 codes, and vice-versa. The intention of these mappings is to offer translations that preserve the complete meaning of the original medical codes being translated.

The upcoming transition to ICD-10 will not affect every type of medical code. ICD-9 diagnosis codes will be replaced by ICD-10 diagnosis codes, ICD-9 procedure codes will be replaced by ICD-10 procedure codes, but CPT/HCPCS codes will remain the same. Therefore, translational crosswalks only exist for ICD-9 diagnosis and procedure codes.

While determining the target codes that would correspond to potential source codes, the CMS attempted to honor the National Library of Medicine (NLM) standard regarding the conversion between coding systems. The NLM believes although it is possible to accurately map from specific concepts to more general concepts, it is impossible to use mappings to add specificity when the original information only addresses general concepts. However, this NLM standard does not supersede the primary purpose of the GEM, which is to provide an acceptable translation for every source system code in both code sets. These mappings do include target system alternatives that are more specific than the source system when better alternatives are not available. Therefore, the crosswalks between ICD-9 and ICD-10 diagnosis and procedure codes contain one-to-one, one-to-many, and many-to-one mappings. Even though non-optimal one-to-many mappings exist, each target code is considered to be an acceptable translation of the source code by the CMS.

Tables 8 and 9 provide sample rows from both the diagnosis and procedure mapping schemes. The Source column contains ICD-9 codes, and the Target column yields the equivalent ICD-10 code(s) for each ICD-9 code. These tables also illustrate the two possible cases that could occur during translation: one-to-one mappings and one-to-many mappings.

Table 8: Example of a One-to-One ICD-9 Procedure Mapping	
ICD-9 Code (Source)	ICD-10 Code (Target)
5283	0FYG0Z2

[4] http://www.cms.gov/Medicare/Coding/ICD10/index.html

Table 9: Example of a One-to-Many ICD-9 Diagnosis Mapping	
ICD-9 Code (Source)	ICD-10 Code (Target)
07989	B338
07989	B341
07989	B342
07989	B344
07989	B348
07989	B9719
07989	B9729
07989	B9789

6.2 General Equivalence Mapping Results

After the GEM crosswalks had been imported in the database, the diagnosis and procedure codes from each individual service were translated from ICD-9 to ICD-10 using basic SQL join statements. The original ICD-9 data from the four selected CCS combinations is described in Table 10. Table 11 contains information regarding the corresponding ICD-10 data that was derived from the Table 10 data. Certain CCS groups, such as 149/84, had little to no one-to-many mappings, while others, such as 205/158, produced a significant number of one-to-many mappings.

Table 10: ICD-9 Pre-General Equivalency Mapping					
CCS Diagnosis	CCS Procedure	Number of Claims	Number of Services	Number of Distinct ICD-10 Diagnosis Codes	Number of Distinct ICD-10 Procedure Codes
203	152	1959	28686	8	2
193	140	990	7172	8	3
205	158	607	7930	26	13
149	84	475	6540	25	3

Table 11: ICD-10 Post-General Equivalency Mapping					
CCS Diagnosis	CCS Procedure	Number of Claims	Number of Services	Number of Distinct ICD-10 Diagnosis Codes	Number of Distinct ICD-10 Procedure Codes
203	152	1959	516807	8	26
193	140	990	71088	8	28
205	158	607	311777	37	313
149	84	475	6540	25	4

7 Logistic Regression

7.1 Logistic Regression Methodology

The statistical process of logistic regression models the relationship between the dependent variable and one or more independent feature variables from the data. Both the fit of the model as well as the significance of the relationships between the dependent and independent variables can be analyzed using this approach. Its ultimate purpose is to estimate the probability of an event occurring, such as the probability a particular claim being fraudulent. However, the dependent variable, predicted using the probability ascertained from the relevant independent variables, is not a precise numerical value. The dependent variable is typically dichotomous in a logistic regressive setting, so the outcome can either be a "1," signaling that the claim is fraudulent for example, or a "0," indicating non-fraudulence.

For this project, a multivariate logistic regression model is used. Let $\pi(x)$ represent the probability of an event that depends on p independent variables. Then, using the inverse logit for modeling the probability:

$$\pi(x) = \frac{e^{\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p}}{1+ e^{\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p}} \tag{5}$$

This form is identical to univariate logistic regression, but there is now more than one independent variable. To obtain the corresponding logit function from this, let X represent the set of covariates $X_1, X_2, \ldots, X_p$ and using basic algebra:

$$logit[\pi(X)] = \ln\left[\frac{\pi(X)}{1-\pi(X)}\right] \tag{6}$$

$$logit[\pi(X)] = \ln\left[\frac{\frac{e^{\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p}}{1+ e^{\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p}}}{1-\frac{e^{\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p}}{1+ e^{\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p}}}\right]$$

$$logit[\pi(X)] = \ln\left[\frac{\frac{e^{\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p}}{1+ e^{\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p}}}{\frac{1}{1+ e^{\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p}}}\right]$$

$$logit[\pi(X)] = \ln[e^{\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p}]$$

$$logit[\pi(X)] = \beta_0 + \beta_1X_1 + \beta_2X_2 + \cdots + \beta_pX_p \tag{7}$$

Just like univariate logistic regression, the probability of a binary event given X is a simple linear function. Equation 5 calculates the probability of an outcome event given the covariate values X_1 through X_p. Using logit transformation to convert the dichotomous outcome, Equation 7 becomes a standard linear regression model. Logit transformation changes the range of $\pi(x)$ from 0 to 1 to $-\infty$ to ∞.

7.2 Logistic Regression Results

The manufacturing of the labeled training data using the anomaly detection process discussed in Section 5 allowed us to use supervised learning methods to predict probable instances of fraud in inpatient claims from 2013. The *glm* function from the R package *stats* was used to construct a logistic regression model for each CCS diagnosis/procedure grouping using the 2012 training data. The appropriate model was then applied to the corresponding ICD-9 data from 2013, and service records with a probability greater than 0.1, or 10%, were flagged. The covariates used within the logistic regression models are listed below.

1. Primary ICD Diagnosis Code
2. Primary ICD Procedure Code
3. Amount Billed
4. Quantity
5. $\sum \frac{\text{Amount Billed}}{\text{Quantity}}$
6. Discharge Status
7. Age
8. Sex

The logit formula uses covariates from the data to calculate the probability of fraud. These independent variables were selected according to the logic of Diagnosis-Related Grouping (DRG), a system of clinical classification that is used by Medicare, Medicaid, and private insurance companies to determine the appropriate cost of inpatient services. DRG labels are assigned through a black box based on the primary ICD-9 diagnosis code, the primary ICD-9 procedure code, the age of the patient, the sex of the patient, the discharge status, and the presence of complications or comorbidities. Fraud detection researchers have previously used DRGs coupled with the amount billed to detect instances of healthcare fraud [19, 22]. Every data attribute contributing to DRG classification, with the exception of the presence of complications, was present in our data. Ideally, a variable indicating complications would have also been used, but the absence of this particular variable did not impact the predictive power of the other DRG-related metrics. These independent variables, along with the cost, quantity, and the subtotal, which was calculated by dividing the cost by the quantity, were selected as the covariates for the logistic regression analysis. The coefficients section of the summary describes each of the covariates that were used in the regression function. In Table 12 below, the diagnosis code, the procedure code, the quantity, the subtotal, the discharge status, the age of the patient, and the sex of the patient were all determined to be statistically significant predictors, according to the p-values provided in the Associated P-Value column.

Table 12: CCS 205/158 ICD-9 Logistic Regression Model			
Covariate	Wald Z-Statistic	Associated P-Value	Significance
Intercept	-5.213	1.86×10^{-7}	***
Diagnosis Code	-10.654	$< 2 \times 10^{-16}$	***
Procedure Code	5.205	1.94×10^{-7}	***
Amount Billed	-0.640	0.52216	
Quantity	-2.957	0.00311	**
Normalized Claim Total	5.951	2.67×10^{-9}	***
Discharge Status	2.038	0.04155	*
Age	-5.918	3.27×10^{-9}	***
Sex	-7.290	3.09×10^{-13}	***

Table 13: CCS 205/158 ICD-10 Logistic Regression Model			
Covariate	Wald Z-Statistic	Associated P-Value	Significance
Intercept	-2.617	0.008858	**
Amount Billed	-0.759	0.447837	
Quantity	-3.406	0.000658	***
Normalized Claim Total	5.411	6.27×10^{-8}	***
Discharge Status	1.754	0.079472	.
Age	-7.376	1.64×10^{-13}	***
Sex	-8.927	$< 2 \times 10^{-16}$	***

These logistic regression models were also used to predict the probability of fraud for the service records in the four CCS groups from the 2013 inpatient claims that had been translated to ICD-10. Since the models were trained using ICD-9 training data, the factor levels for the primary diagnosis code and primary procedure code only contained ICD-9 codes. The models adjusted to the introduction of ICD-10 by dropping the diagnosis code and procedure code as predictors, relying on the remaining six covariates to calculate the probability of fraud.

This logistic regression model was trained using the same CCS 205/158 training data, but eliminated the diagnosis and procedure codes from the formula. The discharge status becomes less statistically significant and the quantity becomes more statistically significant, which can be seen in Table 13. The loss of these two covariates does affect the predicting power and accuracy of the model, and this notion is established in the following subsection.

8 Results

8.1 ICD-9 versus ICD-10

The logistic regression models developed for the ICD-9 testing data used the eight covariates mentioned earlier to predict the probability of service records being fraudulent. Any service with a probability greater than or equal to 0.1 was flagged in the database.

Table : ICD-9 Fraud Results				
CCS Diagnosis	CCS Procedure	Number of Fraudulent Services	Total Number of Services	Percent Fraudulent (%)
203	152	86	28686	0.30
193	140	771	7172	10.75
205	158	453	7930	5.71
149	84	374	6540	5.72

The adjusted regression analysis relied on six covariates to detect possible fraud in the same testing data that was used for the ICD-9 test. Since the CCS classification software uses the same categories for ICD-9 and ICD-10 and every ICD-10 equivalent mapped to the expected CCS group with 100% accuracy, the data was reduced down to the ICD-9 testing data, minus the diagnosis and procedure codes, which eliminated the repetition of services.

Table : ICD-10 Fraud Results				
CCS Diagnosis	CCS Procedure	Number of Fraudulent Services	Total Number of Services	Percent Fraudulent (%)
203	152	85	28686	0.30
193	140	692	7172	9.65
205	158	288	7930	3.63
149	84	210	6540	3.21

Table 6 contains a comparison of the service records flagged as fraudulent using the ICD-9 regression models versus the ICD-10 regression models. For each CCS combination, the ICD-10 regression models, which lacked the diagnosis and procedure covariates, identified less service records as being fraudulent. CCS groups containing a higher number of distinct diagnosis and procedure codes had a more significant discrepancy in the services flagged, while the CCS group 203/152, which contained only 5 distinct diagnosis codes and 1 distinct procedure code, had little discrepancy.

Table : Comparison of Fraud Results					
CCS Diagnosis	CCS Procedure	ICD-9 Fraud Only	ICD-10 Fraud Only	Both Fraud	Neither Fraud
203	152	3	2	83	28598
193	140	131	52	640	6349
205	158	295	130	158	7347
149	84	218	54	156	6112

9 Conclusions

The preliminary results from our experiment indicate that the unavailability of diagnosis and procedure codes as metrics for a supervised and predictive fraud detection system such as a logistic regression model does have an effect on the identification of fraudulent and non-fraudulent inpatient healthcare claims. If a supervised learning method is not trained using labeled ICD-10 data, the predictive power of the diagnosis and procedure codes goes to waste. Even with the presence of a clinical grouper that was compatible for both ICD-9 and ICD-10 codes, the performance of the fraud detection system we implemented still suffered when these two covariates were eliminated.

As the ICD-10 implementation date continues to approach, fraud detection systems that only utilize a small number of metrics or were trained using only ICD-9 data are potentially at risk to experience this transitional impact. However, agencies and companies that have properly prepared their fraud detection systems for this deadline will be able to enjoy the specificity and the predictive power ICD-10 codes provide.

10 Limitations and Future Work

The nature of the data that we used caused various limitations within the project. A lack of known instances of fraud and abuse within the dataset was the most glaring issue with the APCD data. Most research groups exploring the field of healthcare fraud detection work in conjunction with a health insurance company or agency, and this governmental or private entity typically has access to fraudulent claims data. Developing our own fraud detection system without labeled training data proved to be quite challenging, since we were unable to verify our models in any way.

Healthcare data that is available to the public has a significant amount of information redacted or concealed, due to privacy concerns. Researchers have reported a large number of data attributes as valuable features for any fraud detection method, but many of these predictive attributes were not included in the APCD data. Therefore, we were forced to use the basic set of DRG classifiers as the covariates for logistic regression.

Public use data maintained by organizations such as the APCD, the Research Data Assistance Center (ResDAC), the Centers for Medicare and Medicaid Services (CMS), and the Healthcare Cost and Utilization Project (HCUP) better serves research involving healthcare costs, quantity and quality of treatment, morbidity and mortality patterns, and hospital utilization. The sensitive nature of healthcare fraud stemming from legality issues causes any known fraudulent instances to be withheld from publically available claims data. Therefore, within our system, we were flagging claims based on abnormality, rather than abusiveness. This type of healthcare data is quite valuable in a variety of research fields, but its usefulness is limited in the detection and prevention of healthcare fraud.

The system used to store the data and run the experiment was also limiting. Storing, accessing, and manipulating 141 GB worth of data in a local SQL Server database on a desktop computer was less than optimal. The original set of 184 million healthcare service records from

2005 through 2014 had to be reduced down to the 43 million records from 2012 and 2013. Of those 43 million records, approximately 9% were inpatient claims, but R was still unable to consistently handle accessing 3.87 million claims from a database. We ultimately had to settle on using a small subset of inpatient claims from 2012 and 2013 for our experiment.

The natural next step of this project would be to test this idea of a transitional impact using fraud detection systems currently being implemented within the industry. Large health insurance companies and governmental agencies have likely been preparing for this transition to ICD-10 for years, and the necessary adjustments to their fraud detection algorithms and systems have been made. Smaller healthcare insurance entities that rely on simplistic or outdated methods, however, could experience issues regarding the accurate detection of fraud. Testing the capabilities of these smaller or older systems using ICD-10 claims could reveal deficiencies that need to be addressed before the transition to ICD-10 occurs.

11 Appendices

11.1 Sample of Claims Data Format

COVERAGE_CLASS	FROM_YEAR	ADM_YR	DIS_YR	CLAIM_ID_KEY	…
Field Position 1	Field Position 2	Field Position 3	Field Position 4	Field Position 5	….
VARCHAR(3)	VARCHAR(4)	INT(4)	INT(4)	NUMERIC(12)	…

Coverage Class (COVERAGE_CLASS): This field indicated the type of record. For all medical claims records, this value will be MED. Pharmacy Claims are PHM. Dental Claims are DEN.

Date of Service (From) Year (FROM_YEAR): This field contains the date of service of medical claims in a CCYY format. Its source is the Date of Service from element (MC059) in the medical claims.

Admission Year (ADM_YR): This field contains the year of the inpatient admission in CCYY format; Its source is the Admission Date element (MC018) in the medical claims file. These are only populated when valid codes include:

0….Not an inpatient record
-1…Not specified (No discharge date reported)
-2…Not valid (Invalid discharge date code reported)

Discharge Year (DIS_YR): This field contains the year of the inpatient discharge from the hospital in CCYY format; Its source is the Discharge Date element (MC069) in the medical claims file. In addition to dates in CCYY format, valid codes also include:

0….Not an inpatient record
-1…Not specified (No discharge date reported)
-2…Not valid (Invalid discharge date code reported)

Claim Key (CLAIM_ID_KEY): Unique identifier for the claim within the data warehouse.

(This is just a sample of the first five elements of the data. This table continues to include the other 58 remaining attributes.)

11.2 Data Auditing Queries

(Note: The same queries were used for both 2012 and 2013 data subsets.)

```
UPDATE CLAIM_2013
SET MY_KEY = CAST(CONCAT(SUBSTRING(IMPUTED_SERVICE_KEY,1,4), '.',
SUBSTRING(IMPUTED_SERVICE_KEY,6,10)) AS FLOAT);

DELETE FROM CLAIM_2013
WHERE AMT_BILLED <= 0.00 OR QTY <= 0;
```

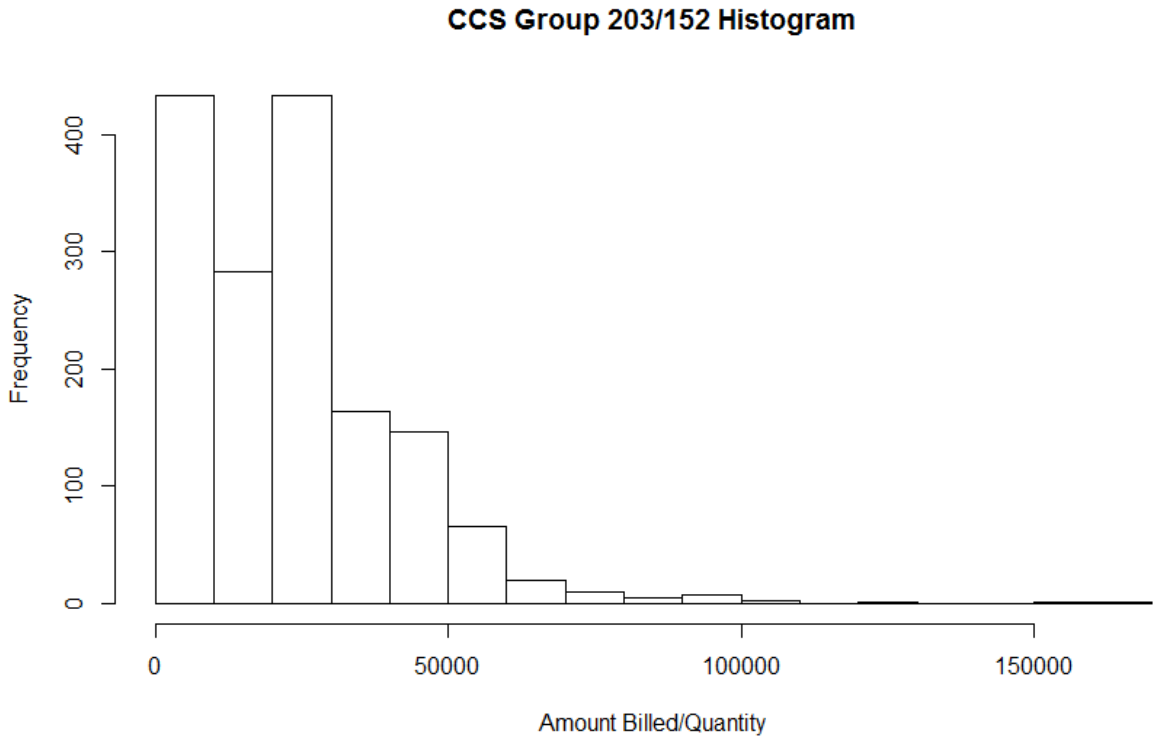
11.3 Clinical Classification Queries

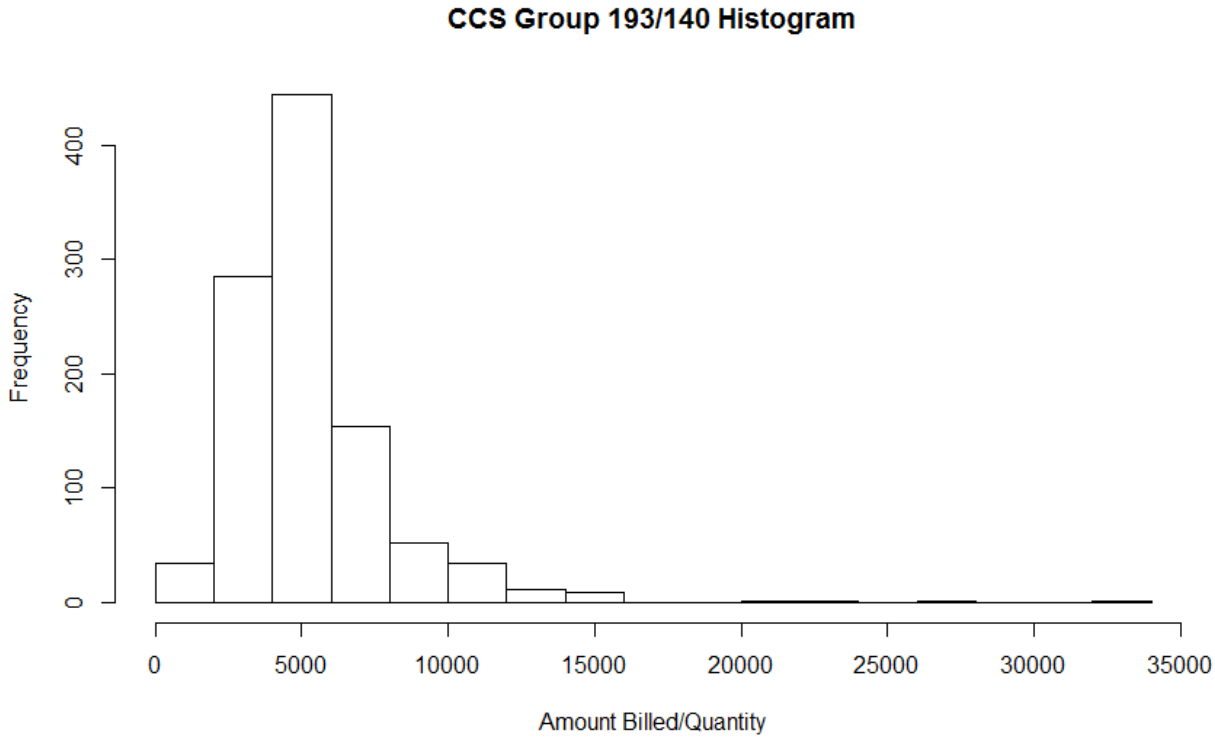
(Note: The same queries were used for both 2012 and 2013 data subsets.)

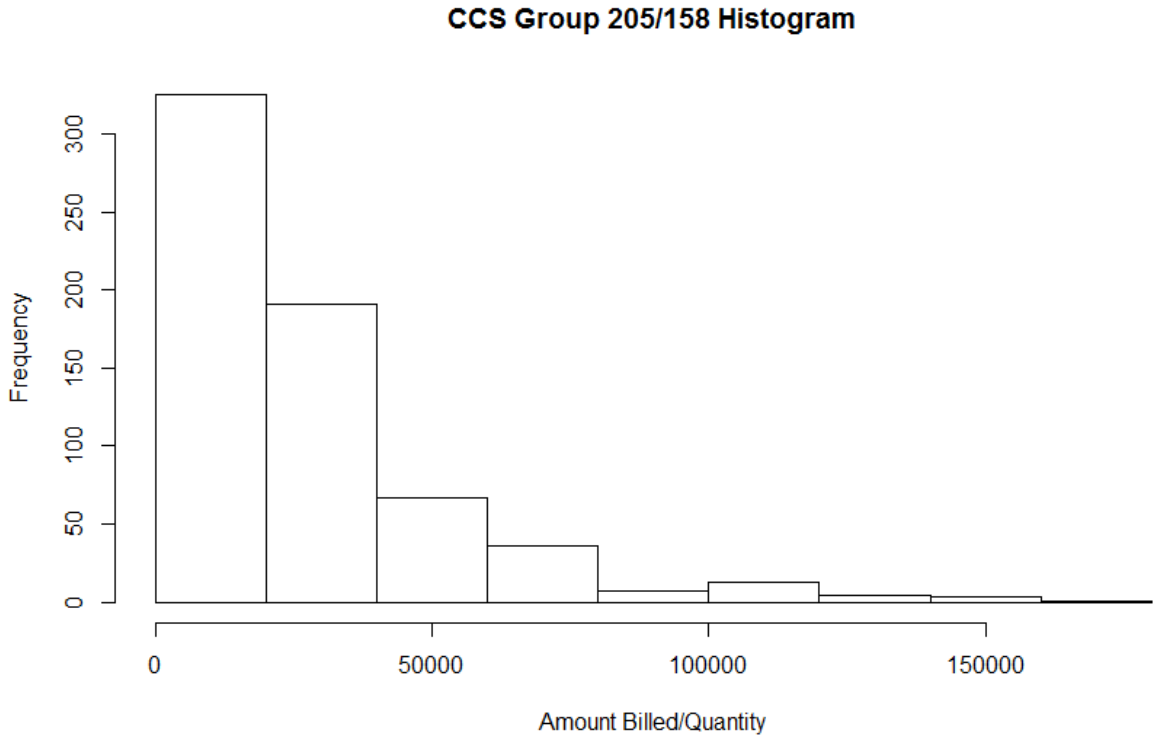
```
UPDATE CLAIM_2013
SET CCS_DX = ccs.CCS
FROM CLAIM_2013 cl INNER JOIN REF_ICD9DX_CCSXW ccs
ON cl.ICD_DIAG_01_PRIMARY = ccs.ICD9_DX
WHERE cl.ICD_DIAG_01_PRIMARY != ''
        AND CCS_DX IS NULL;

UPDATE CLAIM_2013
SET ccs_dx = 'NO_ICD'
WHERE ICD_DIAG_01_PRIMARY = ''
        AND CCS_DX IS NULL;

UPDATE CLAIM_2013
SET ccs_dx = 'INVALID'
WHERE ccs_dx IS NULL;

UPDATE CLAIM_2013
SET CCS_PROC = ccs.CCS
FROM CLAIM_2013 cl INNER JOIN REF_ICD9P_CCSXW ccs
ON cl.ICD_PROC_01_PRI = ccs.ICD9_PROC
WHERE cl.ICD_PROC_01_PRI != ''
        AND CCS_PROC IS NULL;

UPDATE CLAIM_2013
SET ccs_proc = 'NO_ICD'
WHERE ICD_PROC_01_PRI = ''
        AND CCS_PROC IS NULL;

UPDATE CLAIM_2013
SET ccs_proc = 'INVALID'
WHERE ccs_proc IS NULL;
UPDATE CLAIM_2013
SET CCS_CPT = cpt.CCS
FROM CLAIM_2013 cl INNER JOIN REF_CPT_CCSXW cpt
ON cl.proc_code = cpt.cpt
WHERE cl.proc_code != ''
        AND CCS_CPT IS NULL;

UPDATE CLAIM_2013
SET ccs_cpt = 'NO_CPT'
WHERE proc_code = ''
        AND CCS_CPT IS NULL;
```

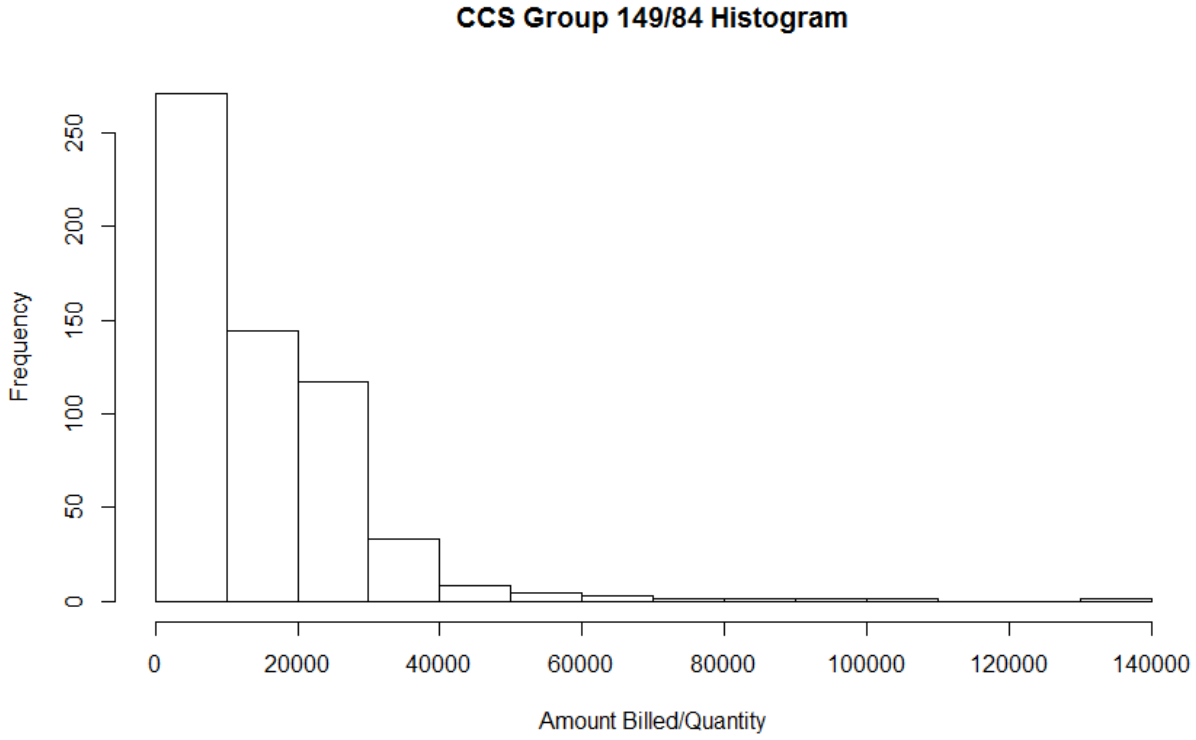
```
UPDATE CLAIM_2013
SET ccs_cpt = 'INVALID'
WHERE ccs_cpt IS NULL;

UPDATE CLAIM_2013
SET CCS_FLAG = 'IGNORE'
WHERE ccs_cpt != 'NO_CPT'
        AND ccs_cpt != 'INVALID'
        AND ccs_proc != 'NO_ICD'
        AND ccs_proc != 'INVALID';

UPDATE CLAIM_2013
SET CCS_FLAG = 'CPT'
WHERE ccs_cpt != 'NO_CPT'
        AND ccs_cpt != 'INVALID'
        AND CCS_FLAG IS NULL;

UPDATE CLAIM_2013
SET CCS_FLAG = 'PROC'
WHERE ccs_proc != 'NO_ICD'
        AND ccs_proc != 'INVALID'
        AND CCS_FLAG IS NULL;

UPDATE CLAIM_2013
SET CCS_FLAG = 'NO_CCS'
WHERE CCS_FLAG IS NULL;

SELECT TOP (10) COUNT(services_key), CCS_DX, CCS_CPT
FROM CLAIM_2012
WHERE CCS_FLAG = 'CPT'
GROUP BY CCS_DX, CCS_CPT
ORDER BY COUNT(services_key) DESC;


SELECT TOP (10) COUNT(services_key), CCS_DX, CCS_PROC
FROM CLAIM_2012
WHERE CCS_FLAG = 'PROC'
GROUP BY CCS_DX, CCS_PROC
ORDER BY COUNT(services_key) DESC;
```

11.4 Distributions of Selected CCS Combinations

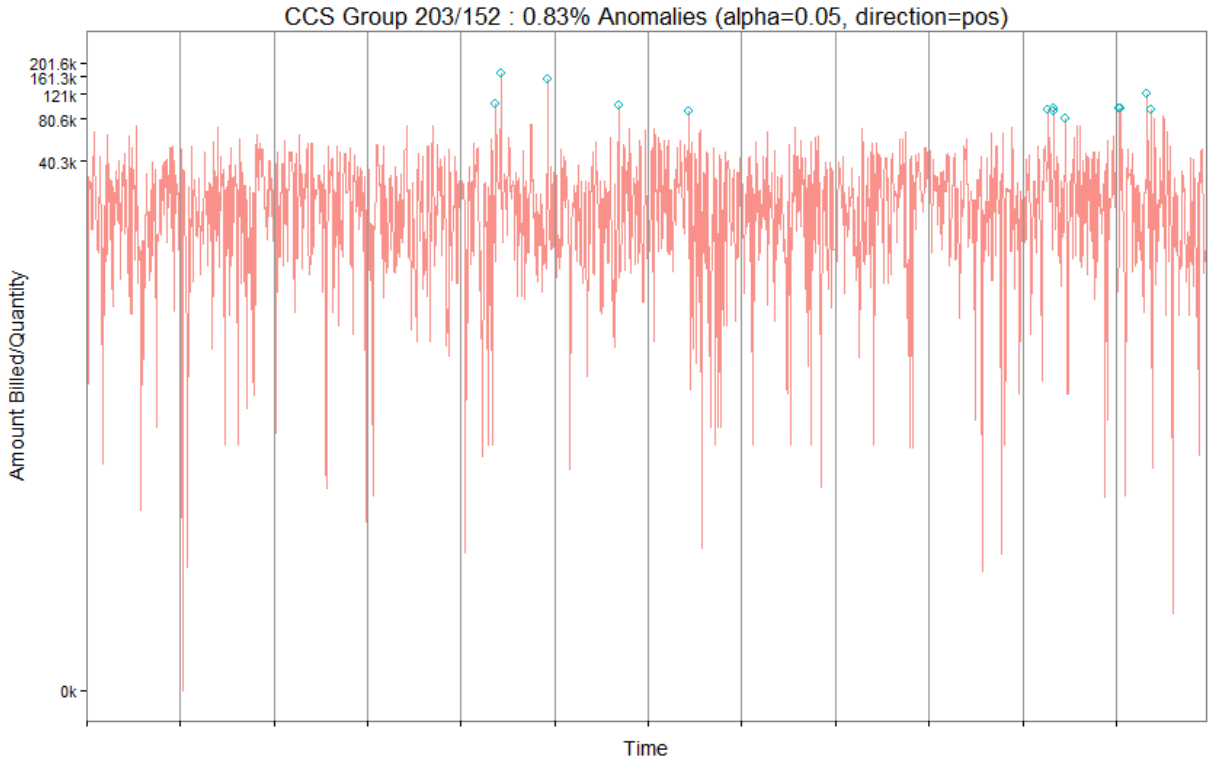


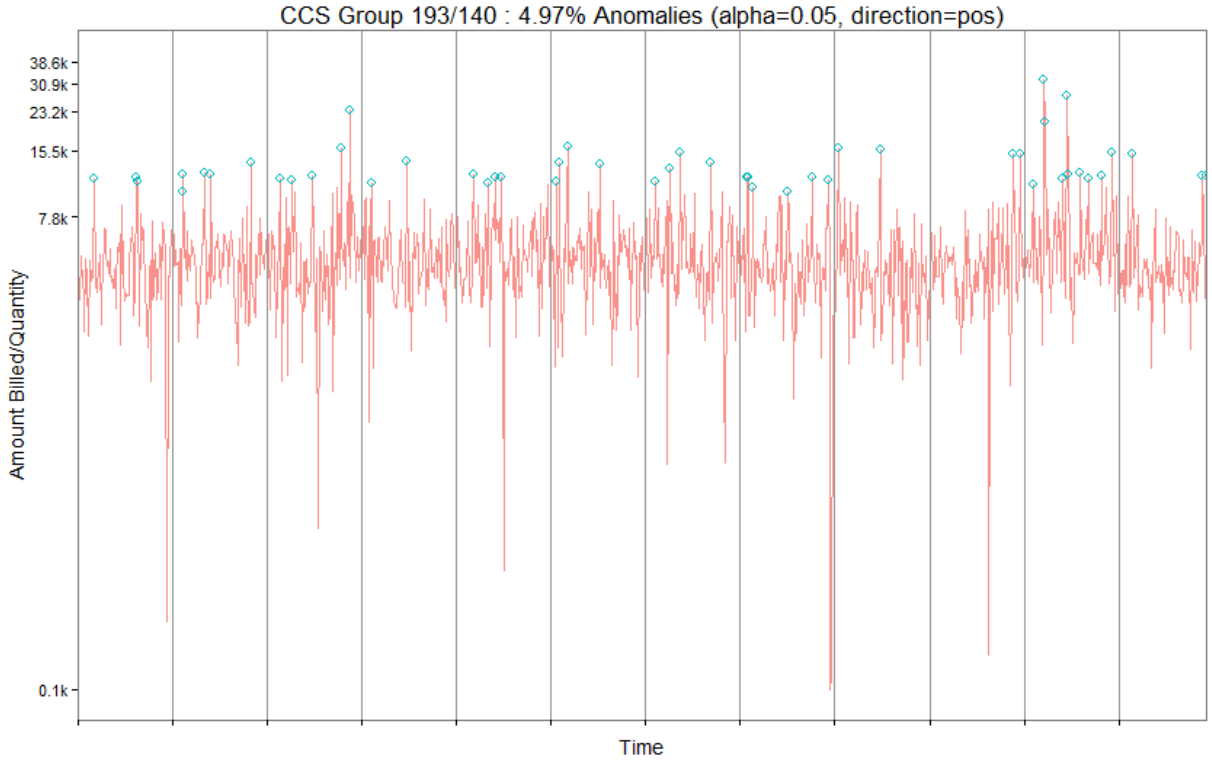


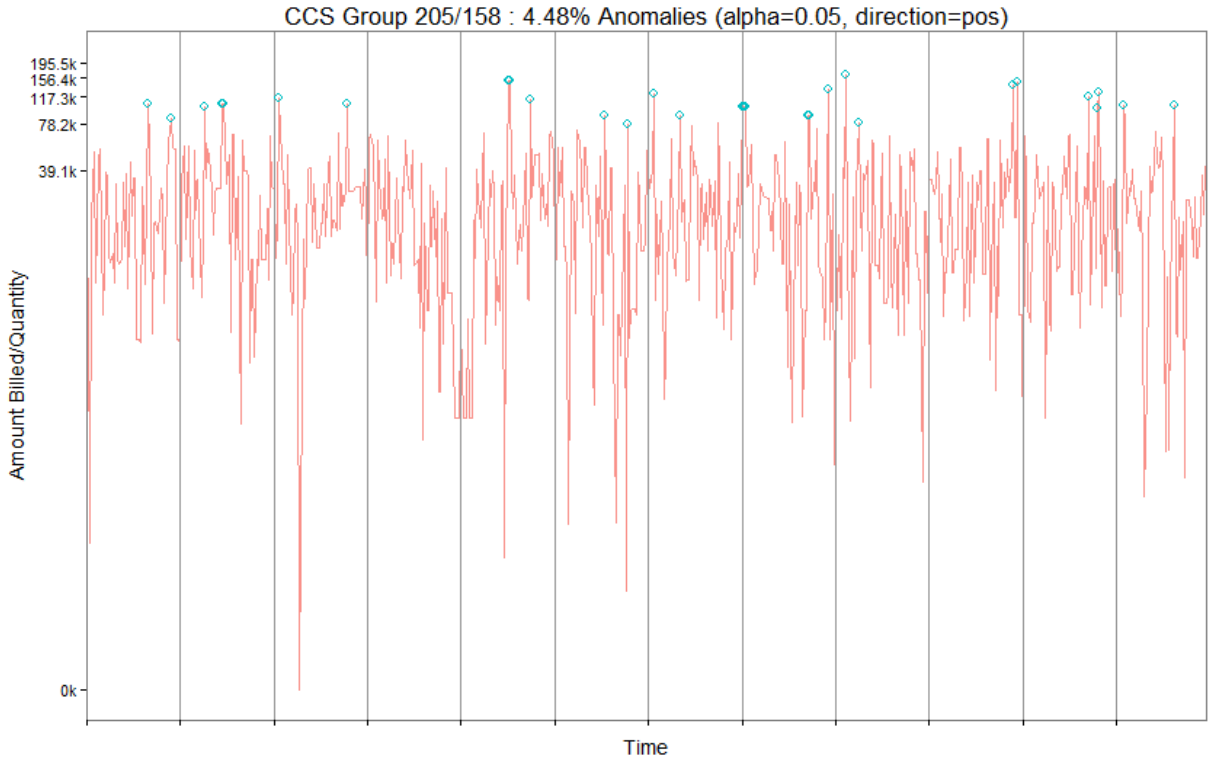

CCS Group 205/158 Histogram
Frequency
0
50
100
150
200
250
300
0
50000
100000
150000
Amount Billed/Quantity

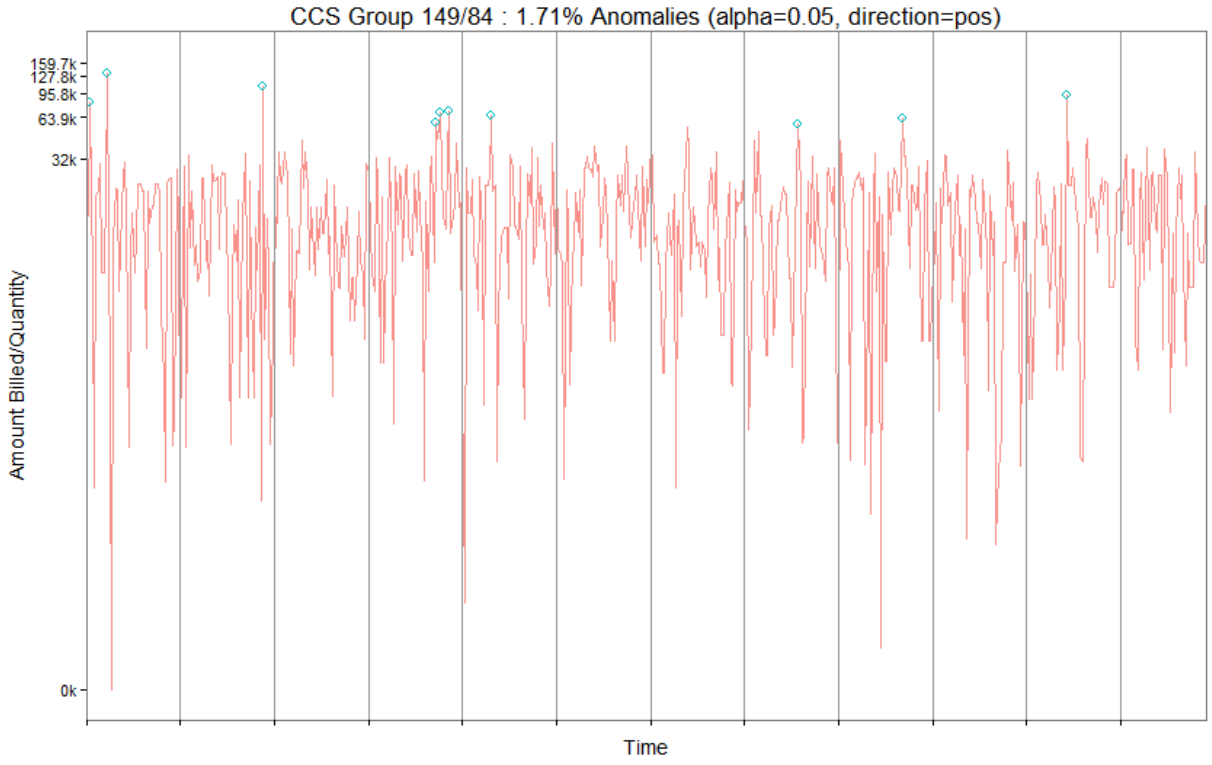

CCS Group 149/84 Histogram
Frequency
0
50
100
150
200
250
0
20000
40000
60000
80000
100000
120000
140000
Amount Billed/Quantity

11.5 Outlier Detection Plots






CCS Group 205/158 : 4.48% Anomalies (alpha=0.05, direction=pos)
195.5k
156.4k
117.3k
78.2k
39.1k
0k
Amount Billed/Quantity
Time


CCS Group 149/84 : 1.71% Anomalies (alpha=0.05, direction=pos)
159.7k
127.8k
95.8k
63.9k
32k
0k
Amount Billed/Quantity
Time

11.6 Outlier Detection Queries and R Code

(Note: The same queries and code were used for all four CCS groups.)

```
SELECT Claim.CCS_DX, Claim.CCS_PROC, Claim.CLAIM_ID_KEY,
Claim.SERVICES_KEY, Claim.ICD_DIAG_01_PRIMARY, Claim.ICD_PROC_01_PRI,
Claim.REV_CODE, Claim.AMT_BILLED, Claim.QTY, Amount.TOTAL, Claim.DIS_STAT,
Claim.AGE, Claim.SEX, Claim.MY_KEY, RowCalc.ROW_NUM
FROM CLAIM_2012 AS Claim, (SELECT CLAIM_ID_KEY,
ROUND(SUM(AMT_BILLED/QTY),2) AS TOTAL FROM CLAIM_2012 WHERE CCS_DX
= '203' AND CCS_PROC = '152' AND CCS_FLAG = 'PROC' AND AMT_BILLED > 0 AND
QTY > 0 GROUP BY CLAIM_ID_KEY) AS Amount,
(SELECT CLAIM_ID_KEY, ROW_NUMBER () OVER (ORDER BY MY_KEY ASC) AS
ROW_NUM FROM CLAIM_2012 WHERE CCS_DX = '203' AND CCS_PROC = '152' AND
CCS_FLAG = 'PROC' AND AMT_BILLED > 0 AND QTY > 0 GROUP BY CLAIM_ID_KEY,
MY_KEY) AS RowCalc
WHERE CCS_DX = '203' AND CCS_PROC = '152' AND CCS_FLAG = 'PROC' AND
AMT_BILLED > 0 AND QTY > 0 AND Claim.CLAIM_ID_KEY = Amount.CLAIM_ID_KEY
AND Claim.CLAIM_ID_KEY = RowCalc.CLAIM_ID_KEY
ORDER BY ROW_NUM;
```

```
SELECT TOTAL
FROM TRAINING
WHERE CCS_DX = '203' AND CCS_PROC = '152'
GROUP BY ROW_NUM, TOTAL
ORDER BY ROW_NUM ASC;
```

```
con <- odbcConnect("HealthcareData")
query1 <- paste0("SELECT TOTAL
                  FROM TRAINING
                  WHERE CCS_DX = '203' AND CCS_PROC = '152'
                  GROUP BY ROW_NUM, TOTAL
         ORDER BY ROW_NUM ASC;")
Results1 <- sqlQuery(con, query1)
Fraud1 <- AnomalyDetectionVec(Results1,
                              max_anoms = 0.05,
                              direction = "pos",
                               alpha = 0.05,
                              period = 131,
                              plot = T,
                              y_log = T,
                              xlabel = "",
                              ylabel = "")
```

```
Fraud1

for (row in 1:nrow(Fraud1$anoms)) {
    query <- paste0(
          "Update TRAINING
          SET TRAINING.FRAUD = 1
          WHERE TRAINING.CCS_DX = '203' AND TRAINING.CCS_PROC = '152' AND
          TRAINING.ROW_NUM = ", Fraud1$anoms$index[row], ";"
)
sqlQuery(con, query)
}

Update TRAINING
SET TRAINING.FRAUD = 0
WHERE TRAINING.FRAUD IS NULL;
```

11.7 Logistic Regression R Code

(Note: The same code was used for all four CCS groups.)

```
query11 <- paste0("SELECT ICD_DIAG_01_PRIMARY AS DIAGNOSIS,
ICD_PROC_01_PRI AS 'PROCEDURE', AMT_BILLED AS COST, QTY AS QUANTITY,
(AMT_BILLED/QTY) AS SUBTOTAL, DIS_STAT AS 'STATUS', AGE, SEX, FRAUD
FROM TRAINING
WHERE CCS_DX = '203' AND CCS_PROC = ‘152';")
Results11 <- sqlQuery(con, query11)

Logit11 <- glm(FRAUD ~ DIAGNOSIS + PROCEDURE + COST + QUANTITY +
SUBTOTAL + STATUS + AGE + SEX, data = Results1, family = "binomial")

query21 <- paste0("SELECT AMT_BILLED AS COST, QTY AS QUANTITY,
(AMT_BILLED/QTY) AS SUBTOTAL, DIS_STAT AS 'STATUS', AGE, SEX, FRAUD
FROM TRAINING
WHERE CCS_DX = '203' AND CCS_PROC = ‘152';")

Results21 <- sqlQuery(con, query21)

Logit21 <- glm(FRAUD ~ COST + QUANTITY + SUBTOTAL + STATUS + AGE + SEX,
data = Results21, family = "binomial")

query12 <- paste0("SELECT SERVICES_KEY, ICD_DIAG_01_PRIMARY AS DIAGNOSIS,
ICD_PROC_01_PRI AS 'PROCEDURE', AMT_BILLED AS COST, QTY AS QUANTITY,
(AMT_BILLED/QTY) AS SUBTOTAL, DIS_STAT AS 'STATUS', AGE, SEX
FROM TESTING
WHERE CCS_DX = '203';")

Predict12 <- sqlQuery(con, query12)

Predict12$fraudICD9 <- predict(Logit11, newdata = Predict12, type = "response")

query22 <- paste0("SELECT SERVICES_KEY, AMT_BILLED AS COST, QTY AS
QUANTITY, (AMT_BILLED/QTY) AS SUBTOTAL, DIS_STAT AS 'STATUS', AGE, SEX
FROM TESTING
WHERE CCS_DX = '203';")

Predict22 <- sqlQuery(con, query22)

Predict22$fraudICD10 <- predict(Logit21, newdata = Predict22, type = "response")
```

11.8 Logistic Regression Models

```
Call:
glm(formula = FRAUD ~ DIAGNOSIS + PROCEDURE + COST + QUANTITY +
    SUBTOTAL + STATUS + AGE + SEX, family = "binomial", data = Results12)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8485  -0.1545  -0.1314  -0.1031   3.6349

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.339e+04  2.802e+05  -0.048 0.961886
DIAGNOSIS   -8.369e-03  4.449e-03  -1.881 0.059945 .
PROCEDURE    1.716e+00  3.437e+01   0.050 0.960187
COST        -6.201e-05  2.846e-05  -2.179 0.029326 *
QUANTITY    -4.557e-03  2.563e-03  -1.778 0.075422 .
SUBTOTAL     1.035e-04  2.855e-05   3.625 0.000289 ***
STATUS      -1.808e-02  6.123e-03  -2.952 0.003156 **
AGE         -4.935e-02  8.979e-03  -5.497 3.87e-08 ***
SEXM        -2.150e-01  1.376e-01  -1.563 0.118099
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2545.4  on 22844  degrees of freedom
Residual deviance: 2420.8  on 22836  degrees of freedom
AIC: 2438.8

Number of Fisher Scoring iterations: 15

Call:
glm(formula = FRAUD ~ DIAGNOSIS + PROCEDURE + COST + QUANTITY +
    SUBTOTAL + STATUS + AGE + SEX, family = "binomial", data = Results13)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2268  -0.3877  -0.3346  -0.2812   2.7713

Coefficients: (1 not defined because of singularities)
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  6.960e+02  4.482e+02   1.553 0.120443
DIAGNOSIS   -1.391e-02  4.639e-03  -2.998 0.002719 **
PROCEDURE    2.990e-02  3.348e-02   0.893 0.371824
COST         4.568e-05  4.843e-05   0.943 0.345582
QUANTITY    -2.779e-04  1.389e-03  -0.200 0.841406
SUBTOTAL     2.128e-04  5.523e-05   3.853 0.000116 ***
STATUS      -1.534e-01  3.116e-02  -4.924 8.50e-07 ***
AGE         -5.050e-02  9.187e-03  -5.497 3.87e-08 ***
SEXTRUE             NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3671.8  on 7639  degrees of freedom
Residual deviance: 3504.7  on 7632  degrees of freedom
AIC: 3520.7

Number of Fisher Scoring iterations: 6
```

```
Call:
glm(formula = FRAUD ~ DIAGNOSIS + PROCEDURE + COST + QUANTITY +
    SUBTOTAL + STATUS + AGE + SEX, family = "binomial", data = Results14)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5594  -0.2887  -0.1670  -0.0971   3.2978

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.595e+02  4.979e+01  -5.213 1.86e-07 ***
DIAGNOSIS   -3.054e-05  2.866e-06 -10.654  < 2e-16 ***
PROCEDURE    3.197e-02  6.142e-03   5.205 1.94e-07 ***
COST        -4.985e-06  7.789e-06  -0.640  0.52216
QUANTITY    -1.348e-02  4.561e-03  -2.957  0.00311 **
SUBTOTAL     5.187e-05  8.716e-06   5.951 2.67e-09 ***
STATUS       1.207e-02  5.920e-03   2.038  0.04155 *
AGE         -3.490e-02  5.897e-03  -5.918 3.27e-09 ***
SEXM        -9.981e-01  1.369e-01  -7.290 3.09e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2629.0  on 8156  degrees of freedom
Residual deviance: 2131.6  on 8148  degrees of freedom
AIC: 2149.6

Number of Fisher Scoring iterations: 8

Call:
glm(formula = FRAUD ~ DIAGNOSIS + PROCEDURE + COST + QUANTITY +
    SUBTOTAL + STATUS + AGE + SEX, family = "binomial", data = Results15)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1746  -0.3173  -0.2433  -0.1854   2.8937

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.996e+02  9.136e+02  -0.219 0.827039
DIAGNOSIS    7.048e-03  1.456e-03   4.842 1.29e-06 ***
PROCEDURE   -4.034e-02  1.790e-01  -0.225 0.821711
COST         6.492e-05  1.882e-05   3.449 0.000562 ***
QUANTITY    -3.132e-03  2.006e-03  -1.561 0.118542
SUBTOTAL    -2.595e-06  2.081e-05  -0.125 0.900720
STATUS      -1.150e-01  4.828e-02  -2.383 0.017180 *
AGE         -3.264e-02  4.275e-03  -7.635 2.26e-14 ***
SEXM        -1.546e-01  1.331e-01  -1.161 0.245577
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2720.8  on 8179  degrees of freedom
Residual deviance: 2464.3  on 8171  degrees of freedom
AIC: 2482.3

Number of Fisher Scoring iterations: 14
```

```
Call:
glm(formula = FRAUD ~ COST + QUANTITY + SUBTOTAL + STATUS + AGE +
    SEX, family = "binomial", data = Results12)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8486  -0.1529  -0.1315  -0.1064   3.6820

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.454e+00  5.111e-01  -2.845 0.004443 **
COST        -6.250e-05  2.851e-05  -2.192 0.028388 *
QUANTITY    -4.544e-03  2.561e-03  -1.775 0.075946 .
SUBTOTAL     1.042e-04  2.860e-05   3.641 0.000271 ***
STATUS      -1.756e-02  6.136e-03  -2.862 0.004206 **
AGE         -4.981e-02  8.856e-03  -5.624 1.86e-08 ***
SEXM        -2.188e-01  1.375e-01  -1.592 0.111484
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2545.4  on 22844  degrees of freedom
Residual deviance: 2426.7  on 22838  degrees of freedom
AIC: 2440.7

Number of Fisher Scoring iterations: 8


Call:
glm(formula = FRAUD ~ COST + QUANTITY + SUBTOTAL + STATUS + AGE +
    SEX, family = "binomial", data = Results13)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1664  -0.3855  -0.3358  -0.2899   2.7688

Coefficients: (1 not defined because of singularities)
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.195e+00  2.762e-01  -4.326 1.52e-05 ***
COST         5.316e-05  4.826e-05   1.102  0.27064
QUANTITY    -1.008e-04  1.379e-03  -0.073  0.94169
SUBTOTAL     2.055e-04  5.507e-05   3.732  0.00019 ***
STATUS      -1.509e-01  3.111e-02  -4.850 1.23e-06 ***
AGE         -5.089e-02  9.224e-03  -5.517 3.45e-08 ***
SEXTRUE             NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3671.8  on 7639  degrees of freedom
Residual deviance: 3519.6  on 7634  degrees of freedom
AIC: 3531.6

Number of Fisher Scoring iterations: 6
```

```
Call:
glm(formula = FRAUD ~ COST + QUANTITY + SUBTOTAL + STATUS + AGE +
    SEX, family = "binomial", data = Results14)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4205  -0.3056  -0.2141  -0.1623   3.2554

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.331e-01  2.801e-01  -2.617 0.008858 **
COST        -5.953e-06  7.843e-06  -0.759 0.447837
QUANTITY    -1.665e-02  4.887e-03  -3.406 0.000658 ***
SUBTOTAL     4.726e-05  8.735e-06   5.411 6.27e-08 ***
STATUS       9.720e-03  5.542e-03   1.754 0.079472 .
AGE         -4.003e-02  5.427e-03  -7.376 1.64e-13 ***
SEXM        -1.181e+00  1.323e-01  -8.927  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2629  on 8156  degrees of freedom
Residual deviance: 2325  on 8150  degrees of freedom
AIC: 2339

Number of Fisher Scoring iterations: 8


Call:
glm(formula = FRAUD ~ COST + QUANTITY + SUBTOTAL + STATUS + AGE +
    SEX, family = "binomial", data = Results15)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0714  -0.3143  -0.2516  -0.1966   2.8645

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.504e+00  1.817e-01  -8.279  < 2e-16 ***
COST         6.168e-05  1.838e-05   3.355 0.000792 ***
QUANTITY    -2.887e-03  1.934e-03  -1.493 0.135538
SUBTOTAL     1.220e-06  2.046e-05   0.060 0.952452
STATUS      -1.273e-01  5.455e-02  -2.334 0.019617 *
AGE         -3.389e-02  4.157e-03  -8.153 3.55e-16 ***
SEXM        -2.855e-01  1.305e-01  -2.188 0.028663 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2720.8  on 8179  degrees of freedom
Residual deviance: 2566.4  on 8173  degrees of freedom
AIC: 2580.4

Number of Fisher Scoring iterations: 8
```

12 Works Cited

[1] Aral, K. D., Güvenir, H. A., Sabuncuoğlu, İ., & Akar, A. R. (2012). A prescription fraud detection model. Computer Methods and Programs in Biomedicine, 106(1), 37-46. http://dx.doi.org/10.1016/j.cmpb.2011.09.003

[2] Bolton, Richard J.; Hand, David J. Statistical Fraud Detection: A Review. Statist. Sci. 17 (2002), no. 3, 235--255. doi:10.1214/ss/1042727940.

[3] Bonchi F, Giannotti F, Mainetto G, Pedreschi D (1999) A classification-based methodology for planning auditing strategies in fraud detection. In Proceedings of SIGKDD99, 175–184

[4] Bresnick, J. (2012, October 31). Healthcare fraud detection faces unknown impact from ICD-10 implementation. Retrieved February 2, 2015, from https://ehrintelligence.com/2012/10/31/healthcare-fraud-detection-faces-unknown-impact-from-icd-10-implementation/

[5] Capelleveen, G. C. (2013). Outlier based predictors for health insurance fraud detection within US Medicaid. Retrieved June 16, 2014 from http://purl.utwente.nl/essays/64417

[6] Chan CL, Lan CH (2001) A data mining technique combining fuzzy sets theory and Bayesian classifier—an application of auditing the health insurance fee. In Proceedings of the International Conference on Artificial Intelligence, 402–408

[7] Copeland, L., Edberg, D., Panorska, A. K., & Wendel, J. (2012). Applying business intelligence concepts to Medicaid claim fraud detection. Journal of Information Systems Applied Research, 5(1), 51.

[8] DeWitt, L., & Baldwin, B. (2013, October 2). Ensuring Provider Payment While Transitioning to ICD-10. Retrieved February 2, 2015, from http://www.bna.com/ensuring-provider-payment-while-transitioning-to-icd-10/

[9] Ekina, T., Leva, F., Ruggeri, F., & Soyer, R. (2013). Application of Bayesian Methods in Detection of Healthcare Fraud. In Chemical Engineering Transaction, 33.

[10] FICO (2013). ICD-10 and Health Care Fraud Detection [White paper]. Retrieved January 20th, 2015 from Healthcare Payer News: http://goo.gl/J37T1v

[11] GAO (1996) Health Care Fraud: Information-Sharing Proposals to Improve Enforcement Effects. Report of United States General Accounting Office

[12] General Dynamics (2012). The Impact of ICD-10 on Fraud, Waste and Abuse Detection [White Paper]. Retrieved January 20th, 2015, from General Dynamics: http://goo.gl/3RykqN

[13] Gutierrez, D. D. (2014, March 31). Data Science and ICD-10 Team Up to Benefit Healthcare. Retrieved February 2, 2015, from http://blog.operasolutions.com/bid/380558/Data-Science-and-ICD-10-Team-Up-to-Benefit-Healthcare

[14] He, H., Graco, W., & Yao, X. (1999). Application of genetic algorithm and k-nearest neighbour method in medical fraud detection. In Simulated Evolution and Learning (pp. 74-81). Springer Berlin Heidelberg.

[15] Hubick KT (1992) Artificial neural networks in Australia. Department of Industry, Technology and Commerce, CPN Publications, Canberra

[16] Hwang SY, Wei CP, Yang WS (2003) Discovery of temporal patterns from process instances. Comp Ind 53:345–364

[17] Kirlidog, M., & Asuk, C. (2012). A Fraud Detection Approach with Data Mining in Health Insurance. Procedia-Social and Behavioral Sciences, 62, 989-994. http://dx.doi.org/10.1016/j.sbspro.2012.09.168

[18] Kliff, S. (2014, February 14). When squirrels attack! There's a medical code for that. Retrieved February 2, 2015, from http://www.washingtonpost.com/blogs/wonkblog/wp/2014/02/14/when-squirrels-attack-theres-a-medical-code-for-that/

[19] Kumar, M., Ghani, R., & Mei, Z. S. (2010, July). Data mining to predict and prevent errors in health insurance claims processing. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 65-74). ACM. http://dx.doi.org/10.1145/1835804.1835816

[20] Li, J., Huang, K. Y., Jin, J., & Shi, J. (2008). A survey on statistical methods for health care fraud detection. Health Care Management Science, 11(3), 275-287. http://dx.doi.org/10.1007/s10729-007-9045-4

[21] Lin J-H, Haug PJ (2006) Data preparation framework for preprocessing clinical data in data mining, AMIA Symposium Proceedings 489–493

[22] Liou, F. M., Tang, Y. C., & Chen, J. Y. (2008). Detecting hospital fraud and claim abuse through diabetic outpatient services. Health Care Management Science, 11(4), 353-358. http://dx.doi.org/10.1007/s10729-008-9054-y

[23] Liu, Q., & Vasarhelyi, M (2013). Healthcare fraud detection: A survey and a clustering model incorporating Geo-location information. In 29th world continuous auditing and reporting symposium (29WCARS). Brisbane, Australia.

[24] Major, J. A., & Riedinger, D. R. (2002). EFD: A Hybrid Knowledge/Statistical Based System for the Detection of Fraud. Journal of Risk and Insurance, 69(3), 309-324.

[25] Musal, R. M. (2010). Two models to investigate Medicare fraud within unsupervised databases. Expert Systems with Applications, 37(12), 8628-8633. http://dx.doi.org/10.1016/j.eswa.2010.06.095

[26] Nigam, S., & Vadlamani, S. (2012). An Innocent Mistake or Intentional Deceit? [White Paper]. Retrieved January 20th, 2015 from HIMSS: http://goo.gl/5BDK5p

[27] Ngufor, C., & Wojtusiak, J. (2013). Unsupervised labeling of data for supervised learning and its application to medical claims prediction. Computer Science, 14(2), 191.

[28] Ortega, P. A., Figueroa, C. J., & Ruz, G. A. (2006). A Medical Claim Fraud/Abuse Detection System based on Data Mining: A Case Study in Chile. DMIN, 6, 26-29. http://dx.doi.org/10.1.1.102.7997

[29] Rockholt, T., Fossey, M., & McLean, M. (2014, January 6). New medical codes can better catch fraud, but training is needed. Retrieved February 2, 2015, from http://www.insurancefraud.org/article.htm?RecID=3305#.VTSoSyHBzRY

[30] SAS (2010). Combating Health Care Fraud [White Paper]. Retrieved January 20th, 2015, from SAS: http://goo.gl/yqib0e

[31] Shan, Y., Jeacocke, D., Murray, D. W., & Sutinen, A. (2008). Mining medical specialist billing patterns for health service management. In J. F. Roddick, J. Li, P. Christen, & P. Kennedy (Eds., pp 105-110), Conferences in Research and Practice in Information Technology, 87. http://dx.doi.org/10.1.1.294.1706

[32] Shan, Y., Murray, D. W., & Sutinen, A. (2009). Discovering inappropriate billings with local density based outlier detection method. In Proceedings of the Eighth Australasian Data Mining Conference-Volume 101 (pp. 93-98). Australian Computer Society, Inc.

[33] Shapiro AF (2002). The merging of neural networks, fuzzy logic, and genetic algorithms. Insurance: Mathematics and Economics 31:115–131

[34] Shin, H., Park, H., Lee, J., & Jhee, W. C. (2012). A scoring model to detect abusive billing patterns in health insurance claims. Expert Systems with Applications, 39(8), 7441-7450. http://dx.doi.org/10.1016/j.eswa.2012.01.105

[35] Sokol L, Garcia B, West M, Rodriguez J, Johnson K (2001) Precursory steps to mining HCFA health care claims. In Proceedings of the 34th Hawaii International Conference on System Sciences

[36] Tang, M., Mendis, B. S. U., Murray, D. W., Hu, Y., & Sutinen, A. (2011, December). Unsupervised fraud detection in Medicare Australia. In Proceedings of the Ninth Australasian Data Mining Conference-Volume 121 (pp. 103-110). Australian Computer Society, Inc.

[37] United States, U.S. Department of Justice, Federal Bureau of Investigation. (2005/2006). Financial Crimes Report To The Public.

[38] Wei CP, Hwang SY, Yang WS (2000) Mining frequent temporal patterns in process databases. Proceedings of international workshop on information technologies and systems, Australia, 175–180

[39] Williams G (1999) Evolutionary Hot Spots data mining: an architecture for exploring for interesting discoveries. Lect Notes Comput Sci 1574:184–193

[40] Williams G, Huang Z (1997) Mining the knowledge mine: The Hot Spots methodology for mining large real world databases. Lect Notes Comput Sci 1342:340–348

[41] Yamanishi K, Takeuchi J, Williams G, Milne P (2004) On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. Data Mining and Knowledge Discovery 8:275–300

[42] Yang, W. S., & Hwang, S. Y. (2006). A process-mining framework for the detection of healthcare fraud and abuse. Expert Systems with Applications, 31(1), 56-68. http://dx.doi.org/10.1016/j.eswa.2005.09.003