Big Data Analytic tools to detect Fraud in healthcare Data

by

Bita Kazemi Zahrani

(Under the Direction of Professor Thiab R. Taha)

Abstract

Healthcare is a billion dollar industry in United States and worldwide. According to Institute of Medicine, almost 30 cents of any medical dollar, is either wasted due to abuse, fraud, or waste, or due to paperwork, unneeded and unnecessary services. Big Data nowadays gives us solutions to analyze, predict and make decisions. The datasets publicly available gives us the opportunity to expose the data to analytic power of Big Data which let us discover new aspects of data. Our goal is to develop a Hadoop Statistics engine, to receive the large public health datasets as an input and deliver a descriptive, more meaningful interpretation of the data to be prepared as an input to predictive and decision engines. The main goal of our engine is to detect anomalies and fraudulent behaviors among our healthcare data according to statistic measures, however it can be further utilized by any type of large dataset as a Big Data Descriptive engine.

Index words: healthcare, Statistical and Descriptive Analytics, Big Data Analytics, Fraud Detection, Hadoop

Big Data Analytic tools to detect Fraud in healthcare Data

by

Bita Kazemi Zahrani

B.Sc., Isfahan University of Technology, 2013

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Master of Science

Athens, Georgia

2015

Big Data Analytic tools to detect Fraud in healthcare Data

by

Bita Kazemi Zahrani

Approved:

Major Professors: Thiab R. Taha

Committee: Hamid R. Arabnia
Lakshmish Ramaswamy

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
December 2015

Big Data Analytic tools to detect Fraud in healthcare Data

Bita Kazemi Zahrani

December 4, 2015

Acknowledgments

I would like to thank cordially to my adviser, Professor Thiab R. Taha, for his patience, guidance and advice, every day, every week of my life as a graduate student. I would like to thank him twice for giving me opportunities to serve the department of computer science as a teaching and research assistant as well as an instructor.

I would like to thank Prof. Hamid R. Arabnia, for his infinite support and brilliant advice and mentor-ship during these two years. If it was not for his advice, I was not here right now.

I would like to thank Prof. Lakshmish Ramaswamy, whom I had the honor to take every course offered by him during these two years. If it was not for his insight, I would have never learned how to conduct research.

I would also to thank my parents, Mahboobeh and Seifollah, for their infinite support throughout my life. Every success I have, I owe to them.

I would also like to thank my life-time friends,Farzad Salimi Jazi and Ladan Yazdanpanah for their support throughout the process of this thesis and my stay in United States from whom I learned to insist on my dreams and try my best toward my goals. I owe this thesis to University of Georgia and my country...Iran.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Big Data and Healthcare

Big Data is a concept introduced in 2001, by Gartner Group in a research to point toward the datasets that are enormously large, in volume, variety and velocity which cannot be managed and processed by traditional data processing application and storage. [1]

In the book, "Big Data and Health Analytics"[2], the authors focus on how we can use every bit of healthcare data and aggregate it as big data concept and help delivering better services to the consumer. Big data does not refer to a big data structure or dataset, but it is more of a solution, potential and tool to manage an organization toward finding an answer to their problems.

The purpose of this thesis is to suggest a small yet viable framework, as a minimum viable product, to exert the power of big data analytics in the area of healthcare especially to the domain of Fraud Detection. Our focus is to apply the state of the art analytical tools to our dataset and investigate the different views and perspectives of our data. We want to study what each of these tools provide to the business and then suggest a framework which can lead us to make more realistic decisions in the area of healthcare. As shown in the figure

Figure 1.1: The landscape of analytics [2]

1.1, big data analytics, can be carried out in four major steps [2]. The first step is reporting stage, when we collect the data, and we have a report of what has happened. The second step is the analyzing stage, where we discuss and mine the data to find out why something has happened. These two first stages are referred to as **Descriptive** stage of analytics. The third stage is to predict what will happen in the future and find the patterns. This is the **Predictive** stage of analytics and on the fourth step we analyze the data for what we should do and take action, regarding the data. This stage is referred to as **Prescriptive** stage. This is the approach and the landscape [2], we will keep in mind and follow throughout the procedural steps of this study.

1.2 Big Data Fraud Detection

The legal definition of Fraud is deliberate deception of information or action to obtain an unfair gain.[3] However in a data and statistics environment, a rare event or an outlier in the data which can induce a financial loss or any financial impact can be presented as a

Fraud. Fraud detection is very difficult from statistics point of view, because as mentioned earlier, the fraudulent events are rare events and behaviours therefore it is not easy to build a predictive model for detecting the further anomalies. From statistical point of view there are techniques such as "long tail", Benford's Law[4], Kurtosis Test[5], Skewness Test [6] and Taleb distribution [7] which can detect the fraud and some of them have been used for even terrorism detection. As an example, Taleb Distribution states that a high probability in small gain shadows a small probability of large loss. Taleb Distribution is the state where huge loss is camouflaged by small gains. Long periods of gains can be interrupted by rare event of large losses and it is easy to defraud the companies by just showing them the partial analysis rather than the full analysis containing the rare events[8]. We use Hadoop to implement Statistical Tests and functions for describing the behavior of different aspects of our dataset. According to[8],Fraud is an illicit practice, illegal act, deception or misrepresentation of data. Fraud can be categorized into five main groups:

1. **Misrepresentation:** Which is falsely presentation, or concealment of some data or facts

2. **Interest Fraud:** which is inducing reliability

3. **Justifiable Reliance:** which is making the manipulated fact, justifiable and reliable

4. **Resulting Damage:** which is manipulating the results in order to achieve a desired fact.

What we aim to perform is to develop a systematic fraud detection methodology with the main focus of detecting any of these frauds autonomically then predict and recognize a fallacious behavior in the data.

1.3 Abuse, Fraud and Waste in Healthcare

Fraud, waste and abuse, contribute to excessive costs in healthcare industry overall expenditure [2]. By Institute of Medicine[13], annually $750 billion dollars which is equal to 30 cents of every medical dollar is either unneeded care, byzantine paperwork or some other source of waste. FBI estimates the total fraudulent expenditure 3-10 percent of the total medical expenses. These unnecessary charges tend to exhaust medical benefits of the patients and prevent them from receiving the needed and appropriate care. The application of analytics to detect anomalies and fraud in the data is not something new, however, applying it to the area of healthcare, especially taking advantage of the Big Data analytics tools, is almost a nascent area of research. By using these tools and paradigms we will be able to investigates million and billions of records and analyze them in a few minutes which makes these techniques favorable and useful as a nationwide, easy to implement methods of analysis.

1.4 Motivation

We have three main motivations throughout the conduct of this study; Financial, Data Analytic Tools and Data Availability.

Financial: The financial motivation beyond this study is that according to [17][12]many of the national debts involve healthcare. Based on the World Health Organization (WHO) [17] 15.2% of US Gross Domestic Product (GDP) is spent on healthcare and this amount will approach 19.5% by the year 2017 [17]. 30% of total healthcare expenditure is marked as waste [12][2] so one of our major motivations is providing and study of analytic tools so that we can contribute to lowering this financial waste.

Data Analytic Tools: Big Data paradigm and its tools has brought a new horizon to the area of analytics. With the state of the art techniques and paradigms we are now able to process huge datasets of billions of records within few minutes. Nowadays many big data

paradigms and frameworks, such as Apache Hadoop[23], SAS[19], SPSS[20], Cloudera[21], Hortonworks[22] are providing scientists and big data analysts with vast pool of big data potentials and facilities. We believe that by proper application and combination of these resources, the future of analytics is and will be revolutionized. One of our major motivations for performing this research is having these tools in hand as a capability to facilitate any big data analysis.

Data Publicity The other major motivation beyond this study is the public dataset available by Center of Medicare and Medicaid Services (CMS) [18][25]. Having the dataset publicly available and used by some researches previously was our last but not least motivation to carry out this study. Currently there are five states, Michigan, Missouri, Oklahoma, California and Louisiana who have signed a consensus to use Data Mining Applications by the Office of Inspector General to prevent Medicaid Fraud. However, we believe that there are many acts that can be applied by using state of the art data analytic techniques to prevent Fraud and detect it at its best place. We do not claim to provide a comprehensive system to analyze the fraud, but we have studied methods which have not been applied before to go one step further toward the development of a National Fraud Detection service for all the Medicare and Medicaid services.

1.5 Organization of Chapters

- **Chapter 1** is dedicated to introductions. The concepts used throughout this study are deeply discussed and described in the first chapter. The motivation and incentives beyond this work are discussed.

- **Chapter 2** describes previously done by researchers on this area of research are further discussed.

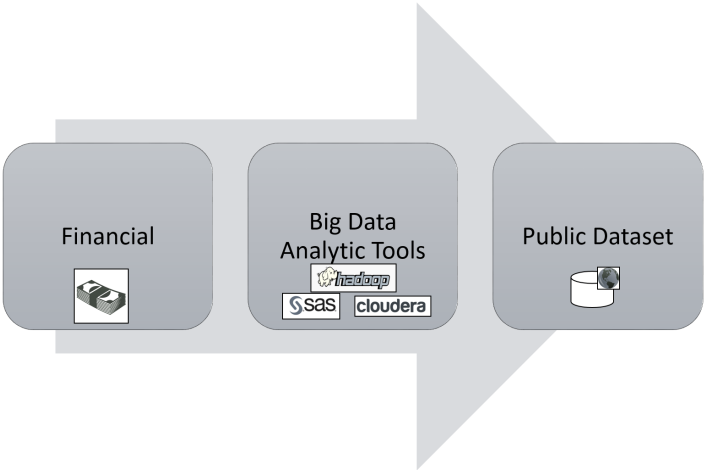

Figure 1.2: Three Motivation Factors for the study

- **Chapter 3** describes the functions, formulas and statistical grounds beyond the methodology.
- **Chapter 4**, is about the dataset, the data processing steps and how inputs and outputs of the system are plugged.
- **Chapter 5**, the results and evaluations from the methodology implemented will be discussed.
- **Chapter 6** we discuss the discoveries and also future work that can be done and the boundary of limitations.
- **Chapter 7** is the bibliography and the references of this work
- **Chapter 8** is the Appendix of the codes implemented for this study.

Chapter 2

Background

2.1 Introduction

There are many steps that have been performed toward design and development of a Fraud Detection System in Healthcare datasets. All these efforts have made major improvements and have provided the ground for further research into the area. We believe that with the vast power of Big Data Analytic tools, we can explore into the area further and obtain better gain from current information and datasets. In this chapter we will review some of the background studies and researches performed prior to this study, their limitations, and their discoveries.

2.2 Knowledge Discovery form Massive Healthcare claims, Oak Ridge National Lab

In [12]from Oak Ridge National Laboratory, researchers have aggregated different types of Healthcare data from Clinical data, Pharmaceutical Research Data, Behavior and Social Sentiment Data and Health Insurance Claims data to provide a fraud detection system.

They have used the claim data, provider enrollment data and Fraudulent providers data from the state of Texas to investigate the claims. They have performed large scale text analytics such as Latent Dirichlet Allocation in order to associate each provider with related topics on the area of specialty and have concluded that not all the providers are associated with their area of expertise which raises the doubt for some anomalies. They have also constructed the provider social network to extract features about closeness of providers, the connectivity and detect anomalies in the social network of Healthcare providers. They have developed Page Rank analysis on the social graph of the providers and also developed a time series statistical analysis to detect fraudulent and non-fraudulent claims.

2.3 Cloudera's Fraud Detection

In an article [8], researchers has discussed the power of Map-Reduce and Hadoop paradigm to detect anomaly and Fraud in financial datasets. They have discussed set of potential Frauds and have mentioned Taleb Distribution [7] in detecting Fraud and detecting rare events.

2.4 Hortonworks Using PageRank to Detect Anomalies and Fraud in Healthcare

In a study performed by Hortonworks [15], the researchers have performed Page Rank on the social network derived from relationship between the healthcare providers and the relationship between healthcare procedures and have detected anomalies regarding the rare and unusual relationships in the graph. In their study they have performed on Medicare Dataset_B, the same dataset we will be using throughout this study, however they have not performed any statistical analytics on their results and have more focused on the graph of

the network. Later they have developed personalized Page Rank in order to detect anomalies in the social graph of their providers. They have used **SociaLite**[24] in order to analyze the graph of connectivity.

2.5 Conclusion

In this chapter, we discussed various works previously performed around application of big data solutions and big data analytics to further analyze healthcare data, or fraud in healthcare data. Our limitation in comparison to some of the previous works, is the dataset. In the study by [12] they have focused on the data from State of Texas, and they have the health insurance data available as well. In the study by Hortonworks[15] they have used the similar dataset however, they have performed no statistical analysis and suggested it as a further addendum, that can be done as a future work to complete their study. We believe that having the public dataset in hand, and statistical analytic tools, we have a good descriptive, yet far away from predictive and prescriptive analytic framework in hand, which can complete any of the previous researches and work in parallel with them.

Chapter 3

Architecture

3.1 Introduction

The aim of this chapter is to explain the mechanism and the design of the system. Our goal is to implement simple and sophisticated statistical formulas to obtain a descriptive view of the current dataset. By descriptive, we mean discovering different aspects of the data, before further moving to a predictive stage. We have developed Skewness Test[6], Kurtosis Test[5],Benford's Law[4], as well as common statistical measures, such as Mean and Standard Deviation. The whole implementation, gives us a statistical Hadoop engine to analyze datasets. The practice is not limited to Fraud Detection and can be used for any data descriptive analysis. The dataset is queried for different attributes and tested with its appropriate functions. We have run the tests in two different formats, first we discuss our dataset based on NPI codes and their related list of CPT codes, which gives us a vector of related practices for each health provider. We can discover how much the provider and the list of services and practices are related to one another. From another perspective, we perform analysis on the list of the allowed charges, submitted charges for each CPT code (the standard codes assigned to each medical procedure) , which gives us the procedure

code along with a vector of the submitted charges. We can run Skewness test, Kurtosis Test, Benford's Law and also Mean, and Standard Deviation on each vector, and calculate how the charges are related . We can detect which procedures have higher potential of committing a fraudulent act. Again we do the same set of tests for each provider and the list of the charges per procedure and we can detect if the provider is overcharging, the health receivers with an abnormal amount of payment. The results of the each test are further visualized and discussed in later sections.

3.2 Steps

There are several standard steps required for a system to perform fraud detection. We can generally categorize the steps in:

1. **Calculation:** in which we perform calculations of statistic metrics such as average, standard deviation, and maximum, minimum and can make a better view and dimension of our data. We perform this step with SAS University Edition tool. We have developed functions to perform the calculation step and report the results.

2. **Classification:** in which we can find patterns and classify our data further into groups. This step can be forwarded to machine learning techniques which are beyond the scope of this work.

3. **Benford's Law:** Based on the reports collected by previous steps, we can apply Benford's Law[4] to detect further anomalies in the data.

    - **Benford's Law**[4] states, that small numbers happen more frequently and with higher probability rather than large numbers. Let's say in a distribution of numbers from 1 to n, the probability that 1 appears is 80% whereas larger numbers

might appear with 5%. The law states that for any digit d from 1 to n, :

$$P(d) = \log_{10}(d+1) - \log_{10}(d) = \log_{10}(\frac{d+1}{d}) \tag{3.1}$$

which means that the probability that each digit happens is the logarithm of its successor digit over itself.

- **Taleb Distribution**[7] mentioned in "Silent risk" should be in bold in economy, there is high probability for small gain and low probability of large loss so we can test the data based on Taleb Distribution based on the expected value
- **Skewness** risk [6] and test
- **Kutrosis** Risk and test

4. **Joining and Relation:** This step is to combine all the reports and results which we have generated from different methods.

5. **Summing:** We perform this step by Hadoop programming.

6. **Validation :** The step can be performed with state of the art Machine Learning techniques.

7. **Visualization:** We pass the results of our calculations to SAS university edition to visualize our data.

We focus on the statistics and descriptive aspect of Big Data solutions and visualization of this descriptive analysis and leave the classification and machine learning analysis of our data, which leads us to a predictive analysis beyond the current study. In the following subsections each of the implemented functions and their interpretations after being applied are discussed with detail.

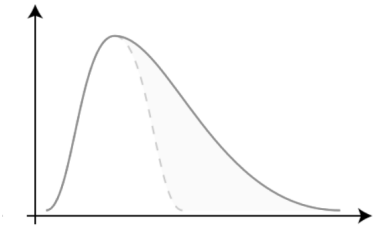

Figure 3.1: Positive Skewness[27].

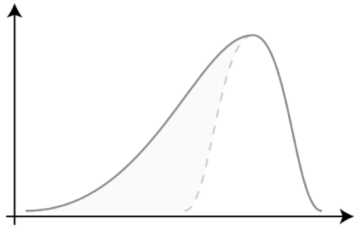

Figure 3.2: Negative Skewness[27].

Skewness

Skewness is a measure of how asymmetric the distribution of probability of a data is. Positive Skewness indicates long tail on the right side of the probability shape however negative value indicates longer tail on the left side. As an example the vector (29, 30 , 31) is evenly distributed, but by adding 20, the vector (20, 29,30, 31) has negative skewness (figure3.2) and by adding 40 to the vector, the vector (29,30,31,40) has a positive skewness (figure 3.1). So we attempt to find these range of skewness for a vector of each procedures provided by a provider who accepts medicare, and then the vector of the amounts charged by the provider. It will give us the shape that each HCPCS_CODE probability distribution has with respect to the amounts allowed and amounts charged. We define the nonparametric skew as :

$$[6]\ Skewness = \frac{\mu - \upsilon}{\sigma} \tag{3.2}$$

to facilitate the computation. In the formula mu is the mean, epsilon is the mode and sigma is the standard deviation of data. We take care of the parts at which mean, mode and median have the same value. In (formula 3.2) we are using Pearson's First Sample Skewness for each vector and have a descriptive view of how the allowed charges vs submitted charges are aligned with their corresponding procedure code.

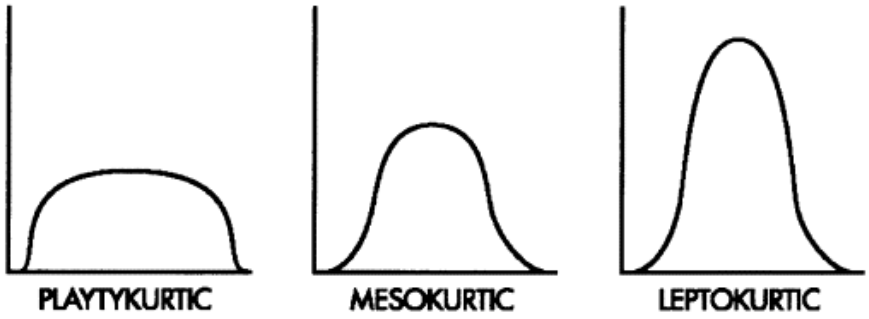

Figure 3.3: platykurtic, mesokurtic and leptokurtic[26]

Kurtosis

Kurtosis: Kurtosis is a statistic function to detect tailedness of a probability distribution of real-valued numerical variables. The function describes the shape of the probability distribution and can be interpreted as "tail weight", 'peakedness" or "lack of shoulders". It acts like the scaled version of the moment of the data. The higher the Kurtosis number, the more variance is happening as a result of infrequent extreme deviations instead of frequent modest deviations. Therefore in our dataset, we can calculate the Kurtosis number regarding each procedure and the charge related to it, with respect to the location and standards and then we can calculate the Kurtosis to see whether the charged amount for the procedure falls into the normal range of Kurtosis or whether it reflects some extreme changes in our dataset. The Kurtosis number for a normal distribution is 3. Thus there is a basis to compare to and interpret our results. If the result is higher than 3, we call the data **platykurtic** (Figure: 3.3), which means they don't have any positive valued tail, as in the uniform distribution. If the Kurtosis number is less than 3, it means that the data is **leptokurtic** (Figure: 3.3), which means that the tail approaching to zero slower than a Gaussian. The function is defined as below:

$$\text{Kurt[X]} = \frac{(\mu)^4}{(\sigma)^4} = \frac{E(X-\mu)^4}{E((X-\mu)^2)^2} \text{ [5]}$$

Method	Description+ Usage
Skewness	Measure of Asymmetry/Symmetry around the mean + Indicator of Positive and Negative deviations around the mean
Benford's Law	Frequent Occurrence of small numbers vs Rare Occurrence of larger numbers + Detect Natural behavior of numbers
Kurtosis	Fourth Momentum around the mean, the tailedness of data + Long Tail Detection
Mean	The average of data + The sum of collection over the count
Standard Deviation	The deviation of data around the mean +Closeness of Data to its mean

Table 3.1: Description and Usage of Statistic Measures

In the formula mu is mean, sigma is the standard deviation of the data.

Mean, Variance, Standard Deviation

We are using this functions to analyze the behavior of our dataset. Each of the functions are defined as below.

Mean :

$$\mu = \frac{\sum_{n=1}^{N} x}{N} \tag{3.3}$$

Standard Deviation :

$$\sigma = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_i - \mu)^2} \tag{3.4}$$

Variance :

$$\sigma^2(variance) = \frac{1}{N}\sum_{n=1}^{N}(x_i - \mu)^2 \tag{3.5}$$

3.3 System Architecture

The System Architecture is displayed in Figure 3.4. The first step is to preprocess the current dataset. The dataset will be explained in details in **Chapter 4** of this report. After cleaning up the data, the dataset is imported to a SAS and MySQL database to facilitate running queries on the data. The results of the queries are imported to text files

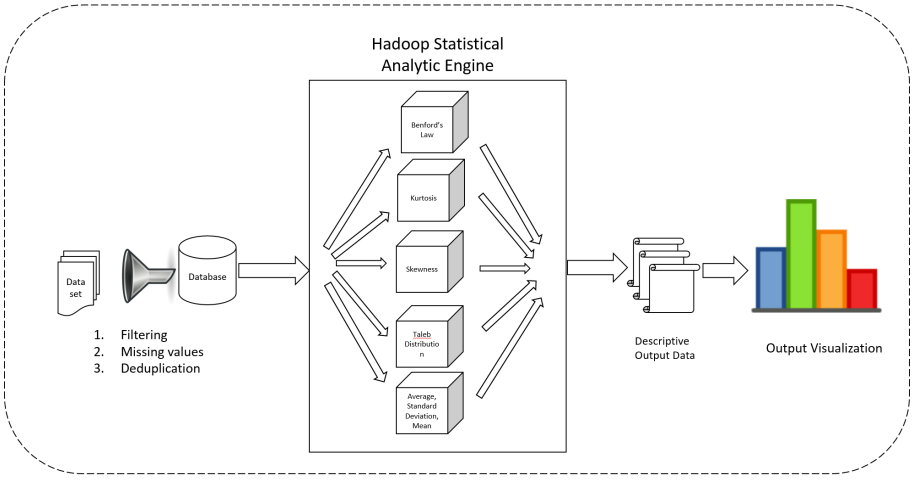

Figure 3.4: The System Architecture in a Nutshell

as an input to their appropriate source codes. We have run Mean, Standard Deviation, Kurtosis, Skewness functions on our data. First we have mapped each provider with a list of its corresponding medical procedures. We have created a vector of each provider and the corresponding procedure codes(CPT or HCPCS_CODE). We have also **mapped** each procedure code to a vector of its allowed charges and have **reduced** the calculation with appropriate function. We have visualized the results of each vector separately and have also applied range queries to partition our data to a unique category of procedures.

3.4 Conclusion

In this chapter we discussed the functions that we have developed and applied to the dataset for exploring different perspectives and aspects of the data. Each function is explained in detail and the usage of the function is described in Table 3.1. We have given a view of the whole architecture of the system to facilitate the explanation of the mechanism by which the system works.

Chapter 4

Data Sources

4.1 Introduction

Medicare Dataset part–B is a part of Obama Administration's effort to provide a transparent, affordable and accountable Healthcare system. The dataset has been made public through Centers for Medicare and Medicaid Services. The part of the dataset we will take into account throughout this study, is the Provider Utilization and Payment Data: Physician and Other Supplier Public Use File known as PUF. The dataset is both available in a text and .sas format. The dataset consists of the attributes listed in on Table 4.1 where each attribute is described.There many more attributes which are beyond the context of this study. In this study, we take into account year 2013 dataset whose data is available on:[25]. A sample part of the dataset is captured in Figure 4.1 and the real names of the providers are replaced by randomly generated data to preserve the privacy.

Attribute	Description
NPI	: National Provider Identifier
HCPCS_CODE	The specific codes for each medical procedure
NPPES_PROVIDER_LAST_NAME	Provider Last Name
NPPES_PROVIDER_FIRST_NAME	Provider First Name
NPPES_PROVIDER_STATE	Provider State
PROVIDER_TYPE	Provider's Specialization
AVERAGE_MEDICARE_ALLOWED_AMT	The Average allowable Amount by Medicare
AVERAGE_SUBMITTED_CHRG_AMT	The Average Submitted Amount
BENE_DAY_SRVC_CNT	Number of Services provided per day

Table 4.1: Important dataset attributes and descriptions

1003089145	Henry S.	Glover	C	M.D.	M	I	1034 N 500 W		PROVO	846043380	
UT US Psychiatry	Y	O	99205	New patient office or other outpatient visit, typically 60 minutes					N		
15	15 15 197.45		0	270	0	140.36	39.317334599				
1003089145	Henry S.	Glover	C	M.D.	M	I	1034 N 500 W		PROVO	846043380	
UT US Psychiatry	Y	O	99214	Established patient office or other outpatient, visit typically 25 minutes					N		
113	30 11 102.78		0	140	0	66.580176991	12.203986798				
1003089145	Henry S.	Glover	C	M.D.	M	I	1034 N 500 W		PROVO	846043380	UT
US	Psychiatry	Y	O	99215	Established patient office or other outpatient, visit typically 40 minutes				N		
32	20	32	137.81	0	188	0	89.6478125	22.579840945			
1003089319	Janet J.	George	J	M.D.	F	I	5301 E GRANT RD	THMEP	TUCSON	85712	AZ
US	Anesthesiology	Y	F	00740	Anesthesia for procedure on gastrointestinal tract using an endoscope						
N	19	18	19	163.39263158	33.432457731	880	150.49916943				
128.77157895	26.214717695										
1003089319	Janet J.	George	J	M.D.	F	I	5301 E GRANT RD	THMEP	TUCSON	85712	AZ
US	Anesthesiology	Y	F	00790	Anesthesia for procedure in upper abdomen including use of an endoscope						
N	13	13	13	346.68692308	97.645833452	1658.8461538	469.28814277				
266.82615385	85.139435637										
1003089319	Janet J.	George	J	M.D.	F	I	5301 E GRANT RD	THMEP	TUCSON	85712	AZ
US	Anesthesiology	Y	F	01480	Anesthesia for procedure on bones of lower leg, ankle and foot	N					
14	14	14	231.84071429	77.740202733	1133.2142857	373.95381277	174.66				
62.335166869											
1003089319	Janet J.	George	J	M.D.	F	I	5301 E GRANT RD	THMEP	TUCSON	85712	AZ
US	Anesthesiology	Y	F	01810	Anesthesia procedure of nerves, muscles, tendons, fascia, and bursae of						
forearm, wrist, or hand	N	17	17	17	118.84882353	19.042974892	592.35294118				
100.58823529	93.291764706		14.94442025								

Figure 4.1: A Glimpse of the Dataset(The names are fakely generated[14])

4.2 Data, attributes and procedures

The dataset HCPCS_CODES are described in Table 4.2. Each Range of codes is related to a range of procedural services. The followings are the main categories. For example the code 103 is related to "Anesthesia for procedure on eyelid" and it falls under the first category.

HCPCS_CODE Ranges	Description
00000-09999	Anesthesia Service
10000-19999	Surgery (Integumentary System)
20000-29999	Surgery (Musculoskeletal System)
30000-39999	Surgery (Respiratory, Cardiovascular, Hemic and Lymphatic Systems)
40000-49999	Surgery (Digestive System)
50000-59999	Surgery (Urinary, Male Genital, Female Genital,Maternity Care and Delivery Systems)
60000-69999	Surgery (Endocrine, Nervous, Eye and Ocular Adnexa, and Auditory Systems)
70000-79999	Radiology Services
80000-89999	Pathology/Laboratory Services
90000-99999	Medicine, Evaluation and Management Services
A0000-V9999	Supplemental Services
0001T-0999T	Category III Code

Table 4.2: HCPCS_Codes and descriptions [28]

4.3 Hadoop and Requirements

We have benefited from Cloudera [21] CDH_5_3. The virtual machine provided by Cloudera runs CentOS 6.4 and we get the VMware version of the machine on a 64-bit host OS. There are a cluster of 3 hadoop nodes and each of them has 4GBs of memory, 4 processors and 70GBs of hard disk .We have SAS University Engine machine for preprocessing the data. The SAS vmware machine has 1GB of memory, 11GBs of hard disk and 2 processors. Hadoop version 2.5 as mentioned on Listing 4.1. We have beforehand, imported the dataset to mysql

database partially, queried each part, and aggregate the results as an input to the Hadoop programs.

```
Hadoop 2.5.0-cdh5.3.0
Subversion http://github.com/cloudera/hadoop -r
    f19097cda2536da1df41ff6713556c8f7284174d
Compiled by jenkins on 2014-12-17T03:05Z
Compiled with protoc 2.5.0
From source with checksum 9c4267e6915cf5bbd4c6e08be54d54e0
This command was run using /usr/lib/hadoop/hadoop-common-2.5.0-cdh5.3.0.jar
```

Listing 4.1: Hadoop Version

4.4 Conclusion

In this chapter, we discussed the dataset, the characteristics of our data, and the features that are important to us throughout the conduct of the study. Each feature of dataset was explained and HCPCS codes were referred to as a guideline for the evaluation chapter. We have performed pre_processing step, removing the duplicates and the missing value of the codes from our dataset. We have run queries on our dataset as well to prepare the text results as an input to the hadoop engine. The pre_processing steps are shown in architecture of the system 3.4.

Chapter 5

Implementation

5.1 Introduction

This chapter is dedicated to the implementation steps and technical aspects of this work implementation. We have harnessed the power of different Big Data techniques in order to process our data and manage it toward the results. We have used SAS, to query dataset, handle the missing values, and run ttests on data. We have used Hadoop Map Reduce in order to implement statistical functions on data and we have used Weka toolkit to find the relations and apply classifiers to the data.

5.2 Map-Reduce

Map-Reduce is a programming model to handle large data sets. The model consists of a map() function, and a reduce() function. The idea beyond map-reduce paradigm is to handle the large dataset with a divide and conquer approach which is breaking the problem into subproblems. The map function, reads the data line by line, and does the filtering and sorting stage. Assume that we have the following data, as the key value pairs: (k1, v1)

(k2, v4)

(k1, v2)

(k3, v6)

(k2, v4)

(k2, v5)

(k3, v6)

The role of map function is to sort and filter the data by map and create a new key value pair for each of the keys. After the map function the data would look like the following :

(k1, v1)

(k1, v2)

(k1, v3)

(k2, v4)

(k2, v5)

(k3, v6)

The map function, does the sorting of the key value pairs, and filters the data by key. There is no aggregation happening in the mapping stage. Shuffling, filtering and sorting happens in map.

The reduce function is where the key value pairs are aggregated by their key. As an example given the map results to the reduce function the results will be aggregated as :

(k1, [v1 , v2, v3])

(k2, [v4, v5])

(k3, [v6])

This is the basic idea beyond mapreduce paradigm. We have used mapreduce to map each of the HCPCS_CODES and prices as :

(procedure_code1, submitted_charge1)

(procedure_code1, submitted_charge2)

(procedure_code2, submitted_charge3)

to a list of submitted charges for each procedure code as the key, and a list of procedures and their allowed charges.

In the reduce function we do the aggregation. We have implemented, mean, Kurtosis, skewness, maximum, standard deviation, and variance in the aggregation phase or the reduce function. Therefor the vector or the list of charges related to each procedure code will be tested for skewness, Kurtosis, maximum, standard deviation and variance and the results of the aggregations and computations are explained in chapter 6, evaluation results.

5.3 SAS

SAS provides a University Edition of their product available to the students.[19] Students can simply enter their university information and sign up to SAS and receive the SAS package at no cost. The deployment of the edition can be done in a few steps. The most important thing is to provide a shared folder to your SAS environment, from which you can read your data and write back to it. We are deploying our programs on SAS Studio which is a browser based programming environment that is connected to a SAS server. We write the code send the code to the SAS server and then the results of the processing will be shown in the browser to our SAS studio session. We have used SAS studio, to run queries on the data for different purposes. As an example the data has many attributes and we want three columns of our data, we have used the code snippet :

```
1 proc import datafile="/folders/myfolders/Medicare_Provider_Util_Payment_PUF_
    CY2013.txt"
2     out=medicare
3     dbms=csv
```

```
4       replace;
5       getnames=yes;
6   run;
7   proc print data = work.second;
8   run;
9   proc sql;
10     create table data as
11        select npi, hcpcs_code,submitted_charge, allowed_charge  from
12           medicare
13              where medicare_provider='y'
14  quit;
```

Listing 5.1: SAS example query

The reason we have used SAS is that conventional databases, such as mysql and sql were sever to handle this dataset however using SAS enabled us to perform query and partition the data the best way possible.

5.4 Weka Toolkit

We have used SAS, to set apart each NPI, HCPCS_CODE and Provider_type for each group of procedures and then we send the results to Weka toolkit to perform some classifications on them. We believe that by performing a classification on the data collected from above, we can test how much and how each procedure code performed in related to the provider_type for which the provider is licensed. We have run J48 decision tree, embedded in Weka toolkit, to get the confusion matrix and classification tree of procedures and provider types for each of the procedure groups. We have used a 66% technique to cross validate. The reason for classification is to evaluate how each provider practice is classified based on their procedure group. The providers whose provider practice are not related to the group of procedure can

be anomalies. As an example if someone is specialized in dermatology but has practiced procedures related to Musculoskeletal group, will raise a source of suspicious and might need further analysis. We have also mapped the provider_type nominal attribute to a numerical attribute, used Kmeans to cluster the HCPCS_CODES based on the providers to see whether the procedures provided fall in the relevant provider_type specialization.

5.5 Conclusion

In this chapter we explained how different tools are used to handle the dataset and lead us toward detecting anomalies and fraud. The further details about the implementation are added in the Appendix section of this work.

Chapter 6

Evaluation Results

6.1 Introduction

We have captured the statistical description and behavior of our dataset in many aspects. One of the aspects we will discuss in the result section is the statistical characteristics of each procedure code (HCPCS_CODE) with respect to the allowed charge amount and submitted charge amount. The Tailedness, skewness, and distribution for each of the range of procedures is displayed in charts and tables. Any outlier residing in the charts, would be a potential for more investigation. The procedural code of the outliers can be mapped to provider lists and investigate more. The functions with a skewness smaller than 3, are on the safe side, however those who lean toward the right, skewness of larger than 3, are potentially prone to overcharging the patient for services.

6.2 Statistical Results & Visualization

We have partitioned data based on 10 different groups of medical procedures as explained in Table 4.2. We have performed data analysis of each group separately. Figure 6.1 is

parameter	Allowed	Submitte
Mean	14.740618	87.263607
Kurtosis	3.0033364	2.1719776
Skewness	1.6441053	1.4397492
Standard Deviation	10.351484	61.476056
Variance	107.15323	3779.3055

Figure 6.1: Anesthesia Services Parameters

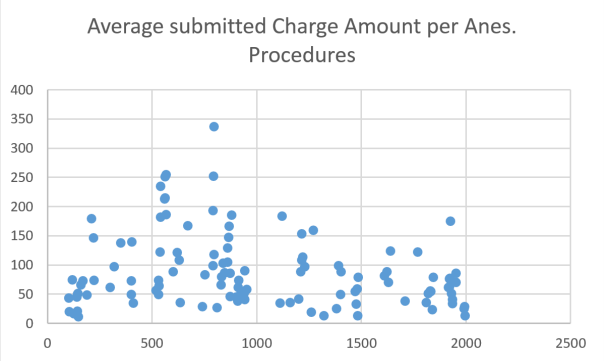

Figure 6.2: Anesthesia Services Distribution of Prices [Horizontal Axis :Procedure Codes, Vertical: Charges per Dollar($)]

related to code 0000-09999 of HCPCS, Anesthesia Services. High difference in mean allowed charge and mean submitted charge is observed. However, characteristics such as Kurtosis and Skewness are very similar which indicates that the distribution of allowed charges and submitted charges have similar characteristics regarding the shape, tailedness and skewness of the descriptor. We observe an outlier in code 796, whose submitted charge is 336.94, but the allowed charge is 58.7. We could mark this as a suspicious point. We go to the original dataset, and query for the code 796, and it can be marked as a potential fraud or waste in dataset. Excluding the point from the results and repeating the analytics will result a different amount for Kurtosis test, 1.53 and 0.88 for allowed and submitted amount correspondingly, however it doesn't affect the skewness of the data. The Kurtosis value is 3.003 for allowed amount, which is similar to normal distribution for allowed charged amount, it is 2.17 for submitted amount and it doesn't indicate anomaly in the general behaviour of submitted charges distribution.

The results in Figure 6.4 are for the medical procedure codes from 10000-19999, Surgery Integumentary System Services. The outlier happens in code 15757. There are three entries in the dataset with that procedure code, we have three pairs of (average_allowed,average_submitted)

parameter	Allowed	Submitte
Average	19.071452	53.552423
Kurtosis	33.409692	21.745749
Skewness	4.935691	4.1514654
Standard Deviation	30.097611	91.726764
Variance	905.86616	8413.7993

Figure 6.3: Integumentary System Services Parameters

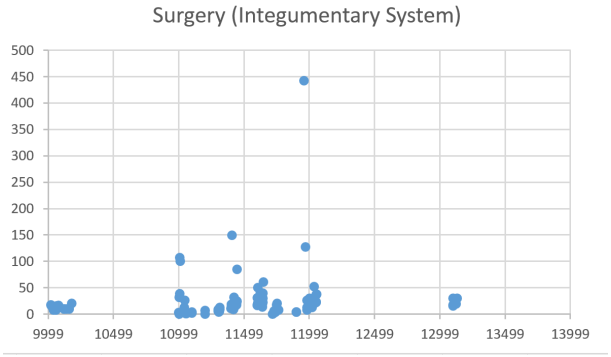

Figure 6.4: Integumentary System Services Distribution of Prices [Horizontal Axis :Procedure Codes, Vertical: Charges per Dollar($)]

with p1= (105.8, 948.11) , p2=(154.05, 666.67) and p3=(155.28,586.36) values. The pair p1 might be a potential point of anomaly in the data. The Kurtosis value indicates that the variance is mostly the result of infrequent extreme deviations in the data rather than frequent moderate ones. The skewness measure for both allowed and submitted charge are almost the same and higher than zero, which indicates that we have almost the same pattern regarding the right-tailedness of our results. Both have the values far above the mean. Excluding the outlier from the data, results in much normal behavior in the submitted charged amount. Even by excluding the outlier, the Kurtosis value still remains 37.71 and 17.51 for allowed and submitted charge correspondingly, which means that the behavior and variance is because of infrequent extreme deviations, but as long as both submitted charge and allowed charge behave in line with one another, we can't suspect the partition for any further anomalies.

Figure 6.6 indicates the results from the partition of data for procedure codes 20000-29999, Surgery Musculoskeletal System Services. The outlier belongs to code 27132, with the pair of(149.69, 773.08) which might be a potential waste. Excluding the outlier reduces the Kurtosis for submitted amount to 0.98 which means that it makes the descriptor more

parameter	Allowed	Submitted
Average	30.50241	130.92151
Kurtosis	5.347432	2.82923347
Skewness	1.914356	1.38223757
Standard Deviation	27.68102	115.955707
Variance	766.239	13445.7261

Figure 6.5: (Musculoskeletal System Services Parameters)

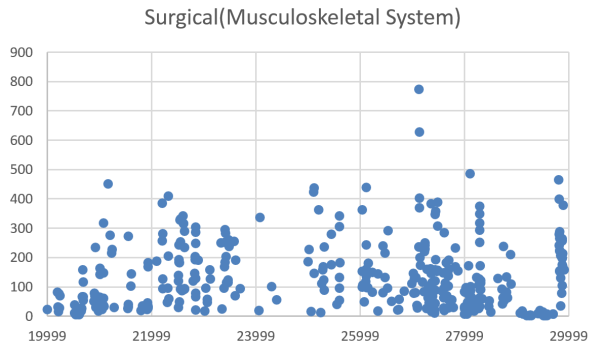

Figure 6.6: Musculoskeletal System Services Distribution of Prices [Horizontal Axis :Procedure Codes, Vertical: Charges per Dollar($)]

of a platykurtic, a flat distribution. The behavior of dataset is not showing extreme changes and the skewness is not severe.

The results in Figure 6.8 show the Surgery Respiratory, Cardiovascular, Hemic and Lympatic System Services, from code 30000 to 39999. There are more than one outlier in the distribution. Excluding the outliers result in Kurtosis of 7, which means the majority of the partition is leptokurtic with infrequent extremes rather than mild frequent deviations. The skewness tends to 2.25 for submitted charges by removing the outliers, however it becomes 3.97 for allowed charges. we analyze the values charged with amounts more than 1000, which are the procedure codes 33979, 38241, 33945, 32853 and 32854 from lowest to highest submitted amount respectively. Retrieving the original data resulting in these anomalies for the corresponding procedural codes results in the table in Figure 6.11 which shows a significant difference between the allowed charged amount and the submitted charged amount.

The values in Figure 6.14 show the distribution of submitted charges over medical procedures from 50000 to 59999, the Urinary, Male Genital, Female Genital, Maternity Care and Delivery System Services. The behavior of allowed charges is very different from that

parameter	Allowed	Submitted
Average	36.310873	142.5038721
Kurtosis	27.4362869	24.0032195
Skewness	4.43690802	4.154525665
Standard Deviation	52.8230954	188.3113115
Variance	2790.27941	35461.15005

Figure 6.7: (Respiratory, Cardiovascular, Hemic and Lymphatic System Services Parameters)

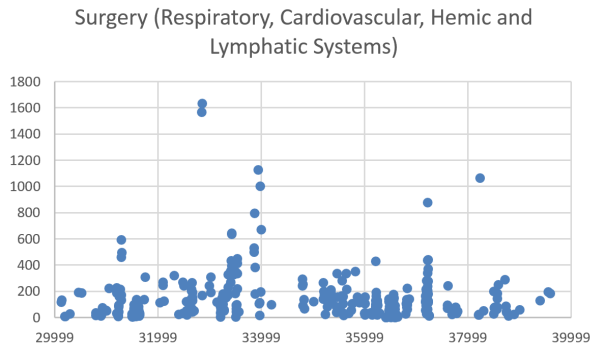

Figure 6.8: Respiratory, Cardiovascular, Hemic and Lymphatic System Services Distribution of Prices [Horizontal Axis :Procedure Codes, Vertical: Charges per Dollar($)]

parameter	Allowed	Submitted
Average	28.58494202	120.3597513
Kurtosis	7.245821843	16.02618552
Skewness	2.427011547	3.516421691
Standard Deviation	34.87772625	161.8979305
Variance	1216.455788	26210.93991

Figure 6.9: Digestive System Parameters

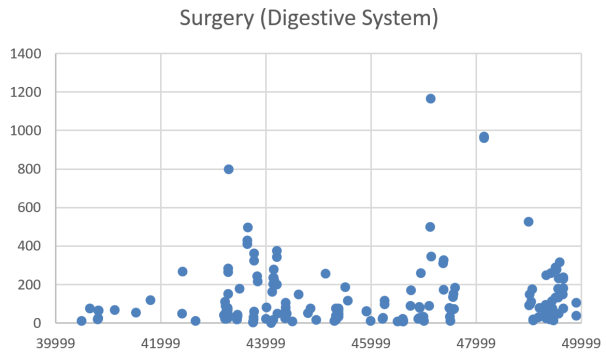

Figure 6.10: Surgery (Digestive System) Services Distribution of Prices [Horizontal Axis :Procedure Codes, Vertical: Charges per Dollar($)]

Row #	HCPCS_CODE	Allowed	Submitted
1	32853	348.67	994.96
2	32853	185.31	577.36
3	32853	364.22	3128.75
4	32854	483.57	1633.6
5	33945	323.84	1124.91
6	33979	107.49	2308.16
7	33979	163.86	891.64
8	33979	110.79	418.15
9	33979	45.64	385.42
10	38241	10.32	2808.23
11	38241	14.99	277.82
12	38241	12.88	103

Figure 6.11: Anomalies in procedure codes 30000 to 39999, the values for Allowed and Submitted column are in Dollar $

Row #	50820	183.93	543.82
1	50820	139.62	1470.86
2	51595	118.54	1250
3	51595	71.47	1299.26
4	51595	182.21	962.82
5	53447	42.72	160.53
6	53447	80.08	1227.27

Figure 6.12: Anomalies in procedure codes 50000 to 59999, the values for Allowed and Submitted column are in Dollar $

of the submitted, which might need extra exploration in our data. We remove the outliers with submitted charge more than 600 and it results in 2.35, 1.63 for Kurtosis and Skewness respectively. The removal of outliers results in the similar pattern of distribution for allowed and submitted charged amount. The outlier procedure codes are 53447 ,50820, 51595. The original data corresponding to the outliers are represented in Figure 6.12. As shown in the Figure 6.12, most of the rows are prone to have anomalies.

The results in 6.10 shows the distribution of submitted charges over procedure codes 40000 to 49999. We put the outlier threshold on the charges over 500 dollars. Excluding the outliers the skewness for both allowed and submitted charges become 1.91 and 1.68 accordingly and the Kurtosis becomes 3.95 and 2.54 respectively. The Skewness is above 0, therefore the data has right-tailedness but the Kurtosis becomes much more similar to a normal distribution. The codes that we analyze further are 48999, 43289, 48150,48153,47135. As results showed 6.15 for these procedures codes there are significant difference between allowed vs the submitted charged amount.

parameter	Allowed	Submitted
Average	28.52735622	121.3649294
Kurtosis	3.187694337	15.79137058
Skewness	1.728627439	3.344630756
Standard Devia	30.43631027	159.7000227
Variance	926.3689829	25504.09723

Figure 6.13: Surgery (Urinary, Male Genital, Female Genital, Maternity Care and Delivery Systems) Parameters

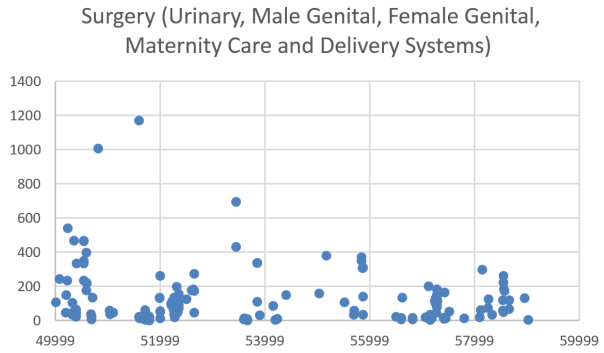

Figure 6.14: Surgery (Urinary, Male Genital, Female Genital, Maternity Care and Delivery Systems) Services Distribution of Prices [Horizontal Axis :Procedure Codes, Vertical: Charges per Dollar($)]

The results in 6.18 shows the distribution of submitted charges over procedure codes 50000 to 59999. We put the outlier threshold on the charges over 1000 dollars. Excluding the outliers, turn the Kurtosis to 2.21,4.11 for allowed and submitted charged amount and turns the Skewness into 1.53,1.89 for allowed and submitted correspondingly. The Kurtosis of the allowed amount is showing lesser frequent changes, and the Kurtosis of the submitted amount on the other hand shows a different behavior of more frequent extremes which might need some more exploration and investigation. The outlier codes are 60820,61595. The Figure 6.17 represent the significant outliers and potential anomalies.

The Figure 6.20 shows the distribution of Radiology submitted charges over the corresponding codes. The only outlier is code 77763. Excluding the outlier makes the Allowed distribution descriptor and the submitted distribution behave pretty much the same. The Kurtosis as shown is very high which is because of extreme infrequencies in the dataset.

Figure 6.22 there are two outliers in codes 81406 and 86833. The Kurtosis and Skewness is very high in both of the allowed and submitted charged amounts which shows similar patterns

Row #	HCPCS_CODE	Allowed	Submitted
1	43289	158.2	1538.5
2	47135	142.9	730.28
3	47135	65.18	323.73
4	47135	196.2	1403.7
5	47135	140.7	691.84
6	47135	294.8	2680.3
7	48150	167.8	1261.2
8	48150	141.3	384.29
9	48150	243.2	1236.1
10	48153	239.3	1218.3
11	48153	209.3	1217.7
12	48999	97.82	527.08

Figure 6.15: Anomalies in procedure codes 40000 to 49999, the values for Allowed and Submitted column are in Dollar $

Row #	HCPCS_CODE	Allowed	Submitted
1	60820	183.93	543.82
2	60820	139.62	1470.86
3	61595	118.54	1250
4	61595	71.47	1299.26
5	61595	182.21	962.82

Figure 6.16: Anomalies in procedure codes 60000 to 69999, the values for Allowed and Submitted column are in Dollar $

parameter	Allowed	Submitt
Average	28.52735622	121.3649
Kurtosis	3.187694337	15.79137
Skewness	1.728627439	3.344631
Standard Deviation	30.43631027	159.7
Variance	926.3689829	25504.1

Figure 6.17: Surgery (Endocrine, Nervous, Eye and Ocular Adnexa, and Auditory Systems) Parameters

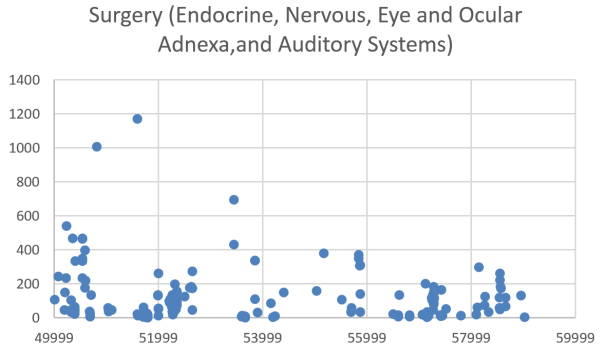

Figure 6.18: Surgery (Endocrine, Nervous, Eye and Ocular Adnexa, and Auditory Systems) Services Distribution of Prices [Horizontal Axis :Procedure Codes, Vertical: Charges per Dollar($)]

parameter	Allowed	Submitted
Average	4.80576634	18.22833781
Kurtosis	10.8295918	14.25830496
Skewness	2.93857648	3.215246969
Standard Deviation	6.06690063	23.47179707
Variance	36.8072833	550.9252576

Figure 6.19: Radiology Services Parameters

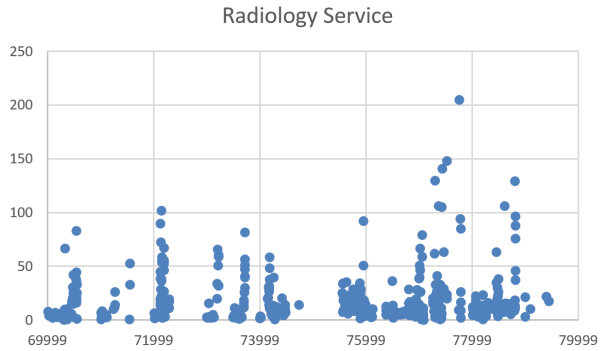

Figure 6.20: Radiology Services Distribution of Prices [Horizontal Axis :Procedure Codes, Vertical: Charges per Dollar($)]

in the nature of both and excluding the outliers smooth out the submitted distribution toward a more normal one. However the data behavior in both has infrequent extremes.

Figure 6.24 there is one outlier on code 93580, and the values allowed and submitted are 87.92 and 441.36 accordingly. Taking out the anomaly, reduce the Kurtosis value to 29.91 and the skewness of submitted to 4.8, but doesn't change that of allowed amount, which means the behavior of both patterns become more similar.

6.3 Classification & Clustering

Running J48, decision tree on data discussed in the implementation chapter, has shown that the data related to group 0 are mostly related to Anesthesiologist Assistants,Anesthesiology,CRNA provider types which seems to be normal. Given the confusion matrix we can see how much the providers are classified into their area of practice given the procedure codes that they have practiced 6.25. We have also mapped each provider_type to a numerical representative and ran Kmeans to cluster the procedure codes and the results of anomalies in irrelevant field of practice is shown in Figure 6.26. The potential anomalies are highlighed in blue.

parameter	Allowed	Submitted
Average	1.231634941	4.218135069
Kurtosis	115.4747259	111.667753
Skewness	9.153395872	8.840152782
Standard Deviation	3.054039739	8.641608838
Variance	9.32715873	74.67740331

Figure 6.21: Pathology/Laboratory Service Parameters

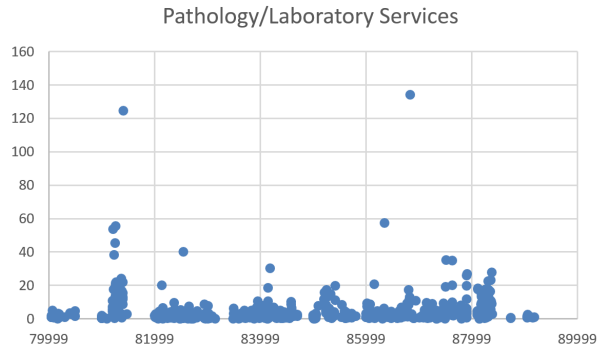

Figure 6.22: Pathology/Laboratory Services Distribution of Prices [Horizontal Axis :Procedure Codes, Vertical: Charges per Dollar($)]

parameter	Allowed	Submitted
Average	4.75244699	14.29183243
Kurtosis	40.9358592	63.76291434
Skewness	5.56481153	6.667169956
Standard Deviation	8.9587728	32.13775705
Variance	80.2596101	1032.835428

Figure 6.23: Medicine, Evaluation and Management Service Parameters

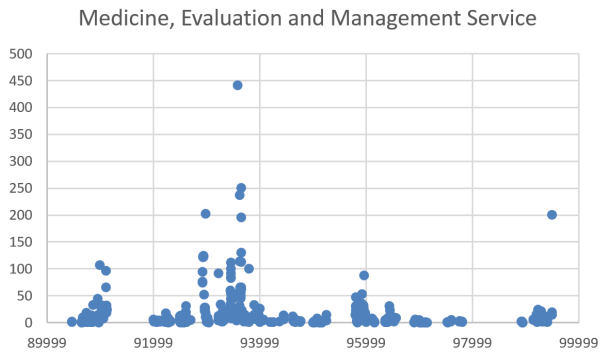

Figure 6.24: Medicine, Evaluation and Management Service Distribution of Prices [Horizontal Axis :Procedure Codes, Vertical: Charges per Dollar($)]

```
 a   b   c   d   e   f   g   h   i   j  <-- classified as
25  31   0   0   0   0   0   0   0   0 |  a = Anesthesiologist Assistants
 0 1039   7   0   0   0   1   0   0   0 |  b = Anesthesiology
 0 391   6   0   0   0   0   0   0   0 |  c = CRNA
 0   1   0   0   0   0   0   0   0   0 |  d = Emergency Medicine
 0   1   0   0   0   0   0   0   0   0 |  e = Family Practice
 0   4   0   0   0   0   0   0   0   0 |  f = Interventional Pain Management
 0   2   0   0   0   0   0   0   0   0 |  g = Oral Surgery (dentists only)
 0   1   0   0   0   0   0   0   0   0 |  h = Critical Care (Intensivists)
 0   1   0   0   0   0   0   0   0   0 |  i = Pain Management
 1   0   0   0   0   0   0   0   0   0 |  j = Allergy/Immunology
```

Figure 6.25: The Confusion Matrix result from running J48 decision tree Weka algorithm on procedure code data from 00000 to 9999 Anesthesia practice

Code group	Description	Provider Classes
0000-9999	Anesthesia Service	Anesthesiology, CRNA
10000-19999	Surgery (Integumentary System)	Dermatology ,Podiatry
20000-29999	Surgery (Musculoskeletal System)	Family Practice, Orthopedic Surgery, Hand and Ambulatory
30000-39999	Surgery (Respiratory, Cardiovascular, Hemic and Lymphatic Systems)	Thoracic, Internal, **Urology, Family Practice**
40000-49999	Surgery (Digestive System)	Diagnostic, Gastroenterology, General, **Vascular**
50000-59999	Surgery (Urinary, Male Genital, Female Genital, Maternity Care and Delivery Systems)	Urology Obstetrics/Gynecology
60000-69999	Surgery (Endocrine, Nervous, Eye and Ocular Adnexa, and Auditory Systems)	Ophthalmology ,Anesthesiology , **Orthopedic**, **Internal Medicine**
70000-79999	Radiology	Diagnostic Radiology Radiation Oncology, Cardiology
80000-89999	pathology	Hematology/Oncology, Internal Medicine, Clinical Laboratory, Family Practice

Figure 6.26: Results of clustering each procedure into provider_type centroids using Kmeans

Chapter 7

Conclusion

In this study, we collected the Medicare_Dataset Part_B, for performing statistical analysis. In the first place we cleaned and preprocessed the data, queried and extracted the attributes that we need to further analyze the data. Then we passed our data to different statistical analysis functions developed by Hadoop and we gained different perspective over different aspects of anomalies in our dataset.

This study, is a point of entry, for those who want to import statistical functionality into the world of big data. The results shows how different views and aspects of data, can be viewed and explained easily by big data functionality.

The results of this study are in a descriptive stage of analytics and can further be enhanced with Machine Learning techniques as a predictive stage and decision making tools for a prescriptive stage of analysis. We believe that there are a vast pool of functions that can be transferred into the Big Data world and can be implemented by Hadoop and other Big Data analysis paradigms and we can build a powerful bridge between Statistics and Big Data world for at least not at last descriptive stage of our analytics.

7.1 Discoveries from Hadoop

We exposed our dataset to the power of Map-Reduce programming and specifically to the statistic functions developed in Map-Reduce paradigm. Working with a dataset of 10 million records, and processing each line, is not feasible with a sequential, conventional programming method, yet the map-reduce paradigm gave us the power to do the analytics in a feasible and fast fashion. We got a descriptive view of our data, which can be fed into Machine Learning, Predictive Analytics and Decision Making modules for more enhanced decision. Our goal was to create a descriptive view of a large dataset by means of map-reduce and big data analytic facilities and statistical functions. We believe that this study can be a starting point into porting the power of map-reduce distributed paradigm into area of healthcare statistical analysis.

7.2 Future Work

The study can be extended to not only healthcare datasets, but to many other datasets to describe the behavior of the data. The descriptive part of analysis can be further improved by expanding the engine and adding more and more functions to it and providing it as a library to the research and academic society. Figure 1.1 can show the future steps. The engine can be embedded into an architecture as a descriptive level and other levels of analytics can be added to the architecture to make a comprehensive healthcare analytic tool. Graph processing capabilities can be further applied to the results and provide us a better vision about the social graph and relatedness of the procedure codes.

Bibliography

[1] Laney, Douglas., 3D Data Management: Controlling Data Volume, Velocity and Variety Gartner. Retrieved 6 February 2001.

[2] Marconi, K., & Lehmann, H. (Eds.). (2014). Big Data and Health Analytics. CRC Press.

[3] Fraud Definition by Wikipedia, https://en.wikipedia.org/wiki/Fraud#Elements_of_fraud

[4] Weisstein, Eric W. "Benford's Law. From MathWorld–A Wolfram Web Resource" http://mathworld.wolfram.com/BenfordsLaw.html

[5] Westfall, P.H. (2014), Kurtosis as Peakedness, 1905 - 2014. R.I.P., The American Statistician 68, 191 - 195

[6] Engineering Statistics Handbook: NISTSEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/

[7] Taleb, N. N. (2014): Silent Risk: Lectures on Probability, Fragility, and Asymmetric Exposures. Fragility, and Asymmetric Exposures (August 5, 2014)

[8] Using Apache Hadoop for Fraud Detection and Prevention : http://blog.cloudera.com/blog/2010/08/hadoop-for-fraud-detection-and-prevention

[9] Kaul, C., Kaul, A., & Verma, S. (2015, March). Comparitive study on healthcare prediction systems using big data. In Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on (pp. 17). IEEE

[10] Zillner, S., Oberkampf, H., Bretschneider, C., Zaveri, A., Faix, W., & Neururer, S. (2014, August). Towards a technology roadmap for big data applications in the healthcare domain. In Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on (pp. 291-296). IEEE.

[11] Mathew, P. S., & Pillai, A. S. (2015, March). Big Data solutions in Healthcare: Problems and perspectives. In Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on (pp. 1-6). IEEE

[12] Chandola, V., Sukumar, S. R., & Schryver, J. C. (2013, August). Knowledge discovery from massive healthcare claims data. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1312-1320). ACM.

[13] Kaiser Health News : www.khn.org

[14] Fake Name Generator Engine: http://www.fakenamegenerator.com/gen-random-us-us.php

[15] Using PageRank to Detect Anomalies and Fraud in Healthcare http://hortonworks.com/blog/using-pagerank-detect-anomalies-fraud-healthcare/

[16] Kentworthy, L. (2011). America's inefficient healthcare system: another look. Consider the Evidence.

[17] World health statistics WHO Library Cataloguing-in-Publication Data, 2011

[18] Center for Medicare and Medicaid Services www.cms.gov

[19] Statistical Analysis System www.sas.com

[20] IBM Predictive Analytic Software www.ibm.com/SPSS_Statistics

[21] Apache Hadoop-based software www.cloudera.com

[22] Open Enterprise Hadoop www.hortonworks.com

[23] Open-source distributed processing framework by Apache http://hadoop.apache.org/

[24] SociaLite: Query language for large-scale graph analysishttp://socialite-lang.github.io/

[25] Dataset : https://www.cms.gov/research-statistics-data-and-systems/statistics-trends-and-reports/medicare-provider-charge-data/physician-and-other-supplier.html

[26] Statistics - Kurtosis (Measure of the "peakedness")_Figure http://gerardnico.com/wiki/data_mining/kurtosis

[27] Skewness Figures : https://en.wikipedia.org/wiki/Skewness#/media/File:Negative_and_positive_skew_diagrams_%28English%29.svg

[28] How to Use The National Correct Coding Initiative (NCCI) Tools by CMS https://www.cms.gov/Outreach-and-Education/Medicare-Learning-Network-MLN/MLNProducts/Downloads/How-To-Use-NCCI-Tools.pdf

Chapter 8

Appendix

8.1 Queries

```
SELECT DISTINCT NPI, hcpcs_code
FROM data
WHERE hcpcs_code <> null;
select  distinct hcpcs_code , hcpcs_description,round(
    average_submitted_chrg_amt/bene_day_srvc_cnt , 2)
from data
where hcpcs_code <>null;
select distinct hcpcs_code,hcpcs_description
from data
where hcpcs_code <>null
```

Listing 8.1: PreProcessing Queries

8.2 vector of HCPCS

```
import java.io.IOException;
import  org.apache.hadoop.util.*;
import org.apache.hadoop.conf.Configuration;
```

```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import java.io.IOException;
import java.util.Iterator;
 import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class Mean {
    public static class  MyMapper extends Mapper<LongWritable, Text, Text,
   Text> {
 String npi;
 String h_code;

 public  void map(LongWritable  key, Text value,
   Context context) throws IOException, InterruptedException {
   String line = value.toString();
   String [] lineSplit = line.split(",");
```

```
34    //finish = line.indexOf(" ");
35    npi= lineSplit[0];
36    h_code = lineSplit[1];
37    context.write(new Text(npi),new Text(h_code));
38
39  }
40 }
41 public static class MyReducer extends Reducer<Text, Text, Text, Text> {
42
43  public  void reduce(Text key, Iterable<Text> values, Context context)
44             throws IOException, InterruptedException {
45   double Count =0.0;
46   double  sum=0.0;
47  double  mean=0.0;
48   String list = "";
49   for(Text  value : values){
50   list += "," . concat(value.toString());
51 }
52   context.write(new Text(key), new Text(list));
53  }
54 }
55
56  public static void main(String[] args) throws Exception {
57         if (args.length != 2) {
58           System.out.println("usage: [input] [output]");
59           System.exit(-1);
60         }
61
62
63         Job job = new Job();
64
```

```
65    job.setJarByClass(List.class);
66    job.setJobName("List");
67    FileInputFormat.addInputPath(job,new Path(args[0]));
68    FileOutputFormat.setOutputPath(job, new Path(args[1]));
69    job.setMapperClass(MyMapper.class);
70    job.setReducerClass(MyReducer.class);
71    job.setOutputKeyClass(Text.class);
72    job.setOutputValueClass(Text.class);
73          System.exit(job.waitForCompletion(true) ? 0 :1);
74
75  }
76
77 }
```

Listing 8.2: Code for related vector of HCPCS

8.3 Mean

```
1  import java.io.IOException;
2  import  org.apache.hadoop.util.*;
3  import org.apache.hadoop.conf.Configuration;
4  import org.apache.hadoop.fs.Path;
5  import org.apache.hadoop.io.*;
6  import org.apache.hadoop.io.Text;
7  import org.apache.hadoop.mapreduce.Job;
8  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
9  import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
10 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
11 import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
12 import java.io.IOException;
13 import java.util.Iterator;
14  import org.apache.hadoop.io.IntWritable;
```

```
15 import org.apache.hadoop.io.Text;
16 import org.apache.hadoop.mapreduce.Reducer;
17 import org.apache.hadoop.io.IntWritable;
18 import org.apache.hadoop.io.Text;
19 import org.apache.hadoop.mapreduce.Mapper;
20 public class Mean {
21     public static class  MeanMapper extends Mapper<LongWritable, Text, Text,
    Text> {
22  String identifier;
23  String elements;
24
25  public  void map(LongWritable  key, Text value,
26    Context context) throws IOException, InterruptedException {
27    String line = value.toString();
28    String [] lineSplit = line.split(" ");
29
30
31
32
33
34    //finish = line.indexOf(" ");
35    identifier= lineSplit[0];
36    elements = lineSplit[1];
37    context.write(new Text(identifier),new Text(elements));
38
39  }
40 }
41 public static class MeanReducer extends Reducer<Text, Text, Text, Text> {
42
43  public  void reduce(Text key, Iterable<Text> values, Context context)
44            throws IOException, InterruptedException {
```

```
45   double Count =0.0;
46   double  sum=0.0;
47  double  mean=0.0;
48   for(Text  value : values){
49   sum += Integer.parseInt(value.toString());
50   Count++;
51 }
52   mean = sum/Count;
53   context.write(new Text(key), new Text(String.valueOf(mean)));
54  }
55 }
56
57  public static void main(String[] args) throws Exception {
58         if (args.length != 2) {
59           System.out.println("usage: [input] [output]");
60           System.exit(-1);
61         }
62
63
64         Job job = new Job();
65   //job.setJobName("Mean");
66   job.setJarByClass(Mean.class);
67   job.setJobName("Mean");
68   FileInputFormat.addInputPath(job,new Path(args[0]));
69   FileOutputFormat.setOutputPath(job, new Path(args[1]));
70   job.setMapperClass(MeanMapper.class);
71   job.setReducerClass(MeanReducer.class);
72   job.setOutputKeyClass(Text.class);
73   job.setOutputValueClass(Text.class);
74         System.exit(job.waitForCompletion(true) ? 0 :1);
75
```

```
 }

}
```

8.4 Skewness Function in Java Hadoop

```


import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.commons.math3.stat.descriptive.moment.FourthMoment;
import org.apache.commons.math3.stat.descriptive.moment.Kurtosis;
import org.apache.commons.math3.stat.descriptive.moment.Mean;
import org.apache.commons.math3.stat.descriptive.moment.Skewness;
import org.apache.commons.math3.stat.descriptive.moment.Variance;

public class SD {

    public static class SkewMapper extends Mapper<Object, Text, Text,
   DoubleWritable> {
```

```
25
26          DoubleWritable val = new DoubleWritable();
27
28          public void map(Object key, Text value, Context context) throws
    InterruptedException, IOException {
29
30              String tokens[] = value.toString().split("\\s+");
31              val.set(Double.parseDouble(tokens[1]));
32              context.write(new Text(tokens[0]), val);
33          }
34      }
35
36      public static class SkewReducer extends Reducer<Text, DoubleWritable, Text
    , DoubleWritable> {
37
38          DoubleWritable result = new DoubleWritable();
39          double count = 0.0;
40
41          public void reduce(Text key, Iterable<DoubleWritable> values, Context
    context) throws IOException, InterruptedException {
42         Vector<Double> mydata = new Vector<Double>;
43              double count = 0.0;
44              double sum = 0.0;
45              double sumSquared = 0.0;
46
47              for (DoubleWritable val : values) {
48                  count += 1.0;
49                  double temp = val.get();
50          mydata.add(temp);
51
52              }
```

```
53          FourthMoment m4 = new FourthMoment();
54          Mean m = new Mean(m4);
55          Variance v = new Variance(m4);
56          Skewness s= new Skewness(m4);
57
58              result.set(s);
59              context.write(key, result);
60          }
61      }
62
63      public static void main(String args[]) throws IOException,
    InterruptedException, ClassNotFoundException {
64
65          Configuration conf = new Configuration();
66
67          if (args.length != 2) {
68              System.err.println("Usage: SD <in> <out>");
69              System.exit(2);
70          }
71
72          Job job = new Job(conf, "calculate skewness");
73
74          job.setJarByClass(Skew.class);
75          job.setMapperClass(SkewMapper.class);
76          job.setReducerClass(SkewReducer.class);
77
78          job.setMapOutputKeyClass(Text.class);
79          job.setMapOutputValueClass(DoubleWritable.class);
80
81          job.setOutputKeyClass(Text.class);
82          job.setOutputValueClass(DoubleWritable.class);
```

```
83
84          FileInputFormat.addInputPath(job, new Path(args[0]));
85          FileOutputFormat.setOutputPath(job, new Path(args[1]));
86          System.exit(job.waitForCompletion(true) ? 0 : 1);
87      }
88  }
```

Listing 8.3: Skewness Function

8.5 Kurtosis Function in Java Hadoop

```
1
2
3  import java.io.IOException;
4
5  import org.apache.hadoop.conf.Configuration;
6  import org.apache.hadoop.fs.Path;
7  import org.apache.hadoop.io.DoubleWritable;
8  import org.apache.hadoop.io.LongWritable;
9  import org.apache.hadoop.io.NullWritable;
10 import org.apache.hadoop.io.Text;
11 import org.apache.hadoop.mapreduce.Job;
12 import org.apache.hadoop.mapreduce.Mapper;
13 import org.apache.hadoop.mapreduce.Reducer;
14 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
15 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
16 import org.apache.commons.math3.stat.descriptive.moment.FourthMoment;
17 import org.apache.commons.math3.stat.descriptive.moment.Kurtosis;
18 import org.apache.commons.math3.stat.descriptive.moment.Mean;
19 import org.apache.commons.math3.stat.descriptive.moment.Skewness;
20 import org.apache.commons.math3.stat.descriptive.moment.Variance;
21
```

```
22 public class SD {
23
24     public static class KurtMapper extends Mapper<Object, Text, Text,
   DoubleWritable> {
25
26         DoubleWritable val = new DoubleWritable();
27
28         public void map(Object key, Text value, Context context) throws
   InterruptedException, IOException {
29
30             String tokens[] = value.toString().split("\\s+");
31             val.set(Double.parseDouble(tokens[1]));
32             context.write(new Text(tokens[0]), val);
33         }
34     }
35
36     public static class KurtReducer extends Reducer<Text, DoubleWritable, Text
   , DoubleWritable> {
37
38         DoubleWritable result = new DoubleWritable();
39         double count = 0.0;
40
41         public void reduce(Text key, Iterable<DoubleWritable> values, Context
   context) throws IOException, InterruptedException {
42        Vector<Double> mydata = new Vector<Double>;
43             double count = 0.0;
44             double sum = 0.0;
45             double sumSquared = 0.0;
46
47             for (DoubleWritable val : values) {
48                 count += 1.0;
```

```
49                  double temp = val.get();
50          mydata.add(temp);
51
52              }
53           FourthMoment m4 = new FourthMoment();
54          Mean m = new Mean(m4);
55          Kurtosis k = new Kurtosis(m4);
56
57              result.set(k);
58              context.write(key, result);
59          }
60      }
61
62      public static void main(String args[]) throws IOException,
    InterruptedException, ClassNotFoundException {
63
64          Configuration conf = new Configuration();
65
66          if (args.length != 2) {
67              System.err.println("Usage: SD <in> <out>");
68              System.exit(2);
69          }
70
71          Job job = new Job(conf, "calculate Kurtosis");
72
73          job.setJarByClass(Kurt.class);
74          job.setMapperClass(KurtMapper.class);
75          job.setReducerClass(KurtReducer.class);
76
77          job.setMapOutputKeyClass(Text.class);
78          job.setMapOutputValueClass(DoubleWritable.class);
```

```
79
80          job.setOutputKeyClass(Text.class);
81          job.setOutputValueClass(DoubleWritable.class);
82
83          FileInputFormat.addInputPath(job, new Path(args[0]));
84          FileOutputFormat.setOutputPath(job, new Path(args[1]));
85          System.exit(job.waitForCompletion(true) ? 0 : 1);
86      }
87  }
```

Listing 8.4: Kurtosis Function

8.6 Loading Data in SAS

```
1  FILENAME REFFILE "/folders/myfolders/Medicare_Provider_Util_Payment_PUF_CY2013
       .txt" TERMSTR=CR;
2
3  PROC IMPORT DATAFILE=REFFILE
4    DBMS=DLM
5    OUT=WORK.IMPORT;
6    GETNAMES=YES;
7  RUN;
8
9  PROC CONTENTS DATA=WORK.IMPORT; RUN;
10
11
12 DATA WORK.IMPORT;
13   LENGTH
14      npi                          $ 10
15      nppes_provider_last_org_name     $ 70
16      nppes_provider_first_name        $ 20
17      nppes_provider_mi            $ 1
```

```
18      nppes_credentials             $ 20
19      nppes_provider_gender         $ 1
20      nppes_entity_code             $ 1
21      nppes_provider_street1          $ 55
22      nppes_provider_street2          $ 55
23      nppes_provider_city           $ 40
24      nppes_provider_zip            $ 20
25      nppes_provider_state          $ 2
26      nppes_provider_country          $ 2
27      provider_type               $ 43
28      medicare_participation_indicator  $ 1
29      place_of_service            $ 1
30      hcpcs_code                  $ 5
31      hcpcs_description             $ 256
32      hcpcs_drug_indicator          $ 1
33      line_srvc_cnt                 8
34      bene_unique_cnt               8
35      bene_day_srvc_cnt             8
36      average_Medicare_allowed_amt      8
37      stdev_Medicare_allowed_amt      8
38      average_submitted_chrg_amt      8
39      stdev_submitted_chrg_amt        8
40      average_Medicare_payment_amt      8
41      stdev_Medicare_payment_amt      8;
42
43  INFILE 'C:\My Documents\Medicare_Provider_Util_Payment_PUF_CY2012.TXT'
44      lrecl=32767
45      dlm='09'x
46      pad missover
47      firstobs = 3
48      dsd;
```

```
49
50 INPUT
51    npi
52    nppes_provider_last_org_name
53    nppes_provider_first_name
54    nppes_provider_mi
55    nppes_credentials
56    nppes_provider_gender
57    nppes_entity_code
58    nppes_provider_street1
59    nppes_provider_street2
60    nppes_provider_city
61    nppes_provider_zip
62    nppes_provider_state
63    nppes_provider_country
64    provider_type
65    medicare_participation_indicator
66    place_of_service
67    hcpcs_code
68    hcpcs_description
69    hcpcs_drug_indicator
70    line_srvc_cnt
71    bene_unique_cnt
72    bene_day_srvc_cnt
73    average_Medicare_allowed_amt
74    stdev_Medicare_allowed_amt
75    average_submitted_chrg_amt
76    stdev_submitted_chrg_amt
77    average_Medicare_payment_amt
78    stdev_Medicare_payment_amt;
79
```

```
LABEL
  npi                    = "National Provider Identifier"
  nppes_provider_last_org_name    = "Last Name/Organization Name of the
 Provider"
  nppes_provider_first_name       = "First Name of the Provider"
  nppes_provider_mi         = "Middle Initial of the Provider"
  nppes_credentials           = "Credentials of the Provider"
  nppes_provider_gender         = "Gender of the Provider"
  nppes_entity_code           = "Entity Type of the Provider"
  nppes_provider_street1        = "Street Address 1 of the Provider"
  nppes_provider_street2        = "Street Address 2 of the Provider"
  nppes_provider_city         = "City of the Provider"
  nppes_provider_zip          = "Zip Code of the Provider"
  nppes_provider_state        = "State Code of the Provider"
  nppes_provider_country        = "Country Code of the Provider"
  provider_type           = "Provider Type of the Provider"
  medicare_participation_indicator  = "Medicare Participation Indicator"
  place_of_service          = "Place of Service"
  hcpcs_code                = "HCPCS Code"
  hcpcs_description           = "HCPCS Description"
  hcpcs_drug_indicator        = "Identifies HCPCS As Drug Included in the
 ASP Drug List"
  line_srvc_cnt             = "Number of Services"
  bene_unique_cnt           = "Number of Medicare Beneficiaries"
  bene_day_srvc_cnt           = "Number of Distinct Medicare Beneficiary/Per
 Day Services"
  average_Medicare_allowed_amt    = "Average Medicare Allowed Amount"
  stdev_Medicare_allowed_amt      = "Standard Deviation of Medicare Allowed
 Amount"
  average_submitted_chrg_amt      = "Average Submitted Charge Amount"
```

```
106     stdev_submitted_chrg_amt       = "Standard Deviation of Submitted Charge
        Amount"
107     average_Medicare_payment_amt   = "Average Medicare Payment Amount"
108     stdev_Medicare_payment_amt     = "Standard Deviation of Medicare Payment
        Amount";
109 RUN;
```