8-2007

Forecasting prescription of medications and cost analysis using time series.

Mussie Angesom Tesfamicael 1975-
University of Louisville

FORECASTING PRESCRIPTION OF MEDICATIONS AND COST ANALYSIS USING TIME SERIES

By

Mussie Angesom Tesfamicael
BSc. in Mathematics, Asmara University, Eritrea
Msc. in Mathematics, Southern Illinois University
M.S.P.H in Biostatistics, University Of Louisville

A Dissertation
Submitted to the Faculty of the
Graduate School of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Mathematics
University of Louisville
Louisville, Kentucky

August 2007

FORECAST PRESCRIPTION OF MEDICATIONS AND COST ANALYSIS USING TIME SERIES

By

Mussie Angesom Tesfamicael

A Dissertation Approved on

August 04, 2007

By the following Thesis Committee:

Dr. Patricia Cerrito
(Thesis Director)

Dr. Adel S Elmaghraby

Dr. Ahmed H Desoky

Dr. Ryan Gill

Dr. Kiseop Lee

DEDICATION

This dissertation is dedicated to my loving parents

Mrs. Rishan Teclebrhan

And

Mr. Angesom Tesfamicael

Without their sacrifice, love, and guidance,

I would not reach my goals and dreams.

ACKOWLEDGMENT

I would like to gratefully acknowledge the enthusiastic and very intelligent supervision of Professor Patricia Cerrito. I am very lucky to have her as my advisor and I could not have imagined having a better advisor and mentor for my dissertation. Her support and guidance during my graduate school years are greatly appreciated.

I also wish to thank all the members of my dissertation committee, Dr. Adel S Elmaghraby, Dr. Ahmed H Desoky, Dr. Ryan Gill and Dr. Kiseop Lee for their advice and assistance with the preparation of this thesis.

Special thanks goes to my brothers, Dr. Aklilu, Tesfaldet, Amanuel, Dawit and lovely sisters Semhar and Senait. They have been a great source of motivation and support. Many thanks to my friends, Dr. Mussie Sium, Dr. Indrias Berhane and Dr. Eshetu Wondmagegnehu for their support and advice when writing my dissertation.

I dedicate this work to my lovely parents Mrs. Rishan Teclebrhan and Mr. Angesom Tesfamicael.

ABSTRACT

FORECASTING PRESCRIPTION OF MEDICATIONS AND COST ANALYSIS USING TIME SERIES ANALYSIS
Mussie Angesom Tesfamicael

August 04 2007

The purpose of this research study is to examine the use of time series forecasting and text mining to investigate the prescription of antibiotics. The specific objective is to examine the relationship between the total payments, private insurance payments, Medicare payments, Medicaid payments, number of prescriptions and quantity of prescriptions for different antibiotics. Currently, there is no method available to forecast antibiotic prescription costs, so we have adopted several methods that will help health care providers and hospitals to know about the prescription of the antibiotics being prescribed. The payment made for each antibiotic is based upon an average cost and total cost that will include the cost of the antibiotics and insurance payments. It will be beneficial to show health care providers the trends of these antibiotics in terms of the cost analysis. It is also beneficial to make comparisons between several antibiotics in terms of the number of prescriptions and to do further study as to why one antibiotic is prescribed more often than others.

We developed time series models that will be used to forecast the prescription practices of the antibiotics. The time series models that we developed for antibiotic prescription are; simple exponential smoothing models, double exponential smoothing model, linear exponential smoothing model. We used exponential models to develop forecasting for antibiotics on which cost increases exponentially. We also developed an autoregressive integrated moving average model for non-stationary data on which the series has no constant mean and variance through time. We developed Generalized Autoregressive Conditional Heteroskedastic Models for volatile variance, and we also incorporated the inflation rate as a model dynamic regressor to see the effect on model forecast. We finally used text mining and clustering to classify the ICD-9 codes into six clusters and make comparisons within each cluster, by plotting the data using kernel density estimation. This project will be beneficial for health care institutions for predicting the trend of the antibiotic prescription, so that further studies can be made why one antibiotic is prescribed more often than others.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

The purpose of this project is to develop time series models to investigate prescribing practices and patient usage of antibiotics with respect to the severity of the patient condition. The cost of antibiotics is rising from year to year; some antibiotics are prescribed more often compared to others even if they have similar properties. It would be of interest to pharmaceutical companies to know the reason for this.

It is the purpose of this project to examine trends and patterns in the prescribing of antibiotic antibiotics. We want to determine whether prescribing patterns change over time, and whether costs change as well. We will develop time series models for each antibiotic to investigate physician practices.

Time series methods have been used to investigate prescribing practices, usually to examine compliance with guidelines.[1] However, these approaches do not yet take into consideration the more recently developed transactional time series methods that can be used with electronic information from the prescription database; for example, information from Pyxis Products (Pyxis Products; San-

Diego, CA) can be downloaded directly and used to investigate prescribing practices.[1,2] One popular time series tool is that of interrupted time series analysis to investigate the impact of an intervention.[3]

The patient condition is defined by a list of ICD9 (International classification of disease) codes developed by the World Health Organization[5]. The ICD9 codes are a series of 5-digit numbers with the first 3 digits representing the main condition and the last 2 digits representing specifics of the condition. For example, “599” represents other disorders of the urethra and urinary tract and “599.55” represents operations on the renal pelvis.

It is the purpose of this research to develop time series models to predict the cost of antibiotics, private insurance payments, Medicaid payments, Medicare payments, the quantity of antibiotics, total payment and to study why the cost is rising in one antibiotic compared to others. We will also investigate how much patients are spending on average for their prescriptions of antibiotics. The data set we have does not take the inflation rate into account, but in this study, we will use the inflation rate as a time-dependent regressor to forecast the cost of antibiotics. Both forecasts, the one that incorporates the inflation rates and the one that does not will be compared. We will compare the cost of antibiotics with respect to the inflation rates.

It is noteworthy that some drugs are called by their generic or brand names. The generic drug name is the chemical name of a drug referring to the chemical makeup of a drug rather than to the advertised brand name under which the drug is sold.[6]

There are over 100 antibiotics in the market, but the majority of them come from only a few types of drugs. The main classes of antibiotics are Penicillin such as penicillin and Amoxicillin, Cephalosporin such as Cephalexin (Keflex), Macrolides such as erythromycin, Clarithromycin and Azithromycin (Zithromax), Fluoroquinolones such as ciprofloxacin (Cipro), Levoflaxacin (Levaquin), and Tetracycline such as Tetracycline and Doxycycline (Vibramycin). In this project, we will study the cost analysis of these antibiotics in relation to time [9].

The data set was obtained from the Medical Expenditure Panel Survey (MEPS) on prescribed antibiotics, using data from 1996-2004, with 2004 the most current year posted. The data for each year is contained in a separate dataset, for instance, the prescribed antibiotics for the year 1996 is on file HC-010A; this file contains the patient's antibiotic information and cost for the drugs.[4] The other datasets also contain information for one of the years, 1997-2004. Information is merged for all the years to have a single dataset. The preprocessing is described in detail in chapter 2.

Antibiotics are among the most frequently prescribed medications nowadays. Antibiotics cure disease by killing or injuring bacteria. The first discovered antibiotic was penicillin, which was discovered from a mold culture. Today, over 100 different antibiotics are available to doctors to cure minor discomforts as well as life-threatening infections.

Antibiotics only treat bacterial infection, although they are used in a wide variety of illnesses. Antibiotics don't cure viral infections such as the common cold; nor can they treat fungal infections[9]. Most antibiotics have two names, a brand name created by the drug company that manufactures the drug and a generic name based on the chemical composition of the drug.

Previous studies

Prescription drugs account for 19.88 percent of total health care expenditures for the U.S. civilian, non-institutionalized population for the years 2003 and 2004. The prescription expenditure was 11.9 percent of total health expenditures in 1996. The increase of the overall costs of health care has a direct effect on the increase of the cost of prescription drugs; as a result, more attention has been given to studying these costs. The percentage of total health care expenditures for prescription drugs increased from 11.5 percent in 1996 to 19.5 percent in both 2003 and 2004 for persons under the age of 65 [7].

The average expense per purchase for brand name and generic medications has increased over the years, 1999 to 2003, with the average expense for generic medication rising from $23.48 to $33.53 and the average expense for a brand name drug rising from $59.49 to $82.53 [8].

Previous studies have not developed time series models to investigate the cost and prescription practices for medications. This research will also define a measure to classify the severity of a patient's condition using ICD9 codes.

Purpose of the Study

The main purpose of this study is to develop time series models to forecast the cost of antibiotics and to classify patient usage of antibiotics with respect to patient conditions using text-mining clustering. We will also study on average how much patients are spending on antibiotics.

Another purpose of this research is the changing behavior of the cost of antibiotics by introducing intervention variables. We will analyze these data using interrupted time series analysis. We also examine the payments made for Medicare and Medicaid and study how much patients are spending on average for prescriptions. We also want to clear that we have only studied the antibiotic prescription of the several medications that are available in the market.

In this dissertation, we develop time series models, such as ARIMA models, ESM (Exponential Smoothing Model), and the Heteroskedastic models (ARCH and GARCH). Autoregressive procedures are used to develop a model for each antibiotic so that cost can be forecast. We also use data mining techniques, in particular, text mining to classify the prescriptions based on the patient's severity conditions using ICD9 codes. Text mining is used to classify observations based on the content of the words listed in each variable with the expectation maximization algorithm (EM).

This dissertation is organized as follows: Chapter two gives detailed information about how we preprocessed the data. Chapter three introduces the Exponential smoothing models (ESM). ESM starts with an infinite past and applies a weighted average with weights that exponentially decay to zero. Chapter four describes in detail the theory of ARIMA models. Chapter five is about Heteroskedastic models. It analyzes and forecasts for volatile time series data. Chapter six describes the theory of text mining and classification. Chapter seven shows the results of our analyses through several different models. Statistical data analysis and figures will be used throughout to determine the best model. Chapter eight gives the summary, conclusion and recommended follow-up studies.

CHAPTER 2

DATA PROCESSING

In this chapter, we give a detailed discussion of how the data were preprocessed before a model for prediction was built. Most studies that involve simulation do not require preprocessing. The researcher has to generate random data and build a specific model that fits the data generated. However, in time series analysis, one has to preprocess the data before a model for prediction is built. In particular, in this project, a large amount of time was required to preprocess the data. The data set for this project was collected from the medical expenditure panel survey (MEPS) [4]. The MEPS contains new and extensive data on the use of health services and health care in the United States. MEPS is conducted to provide nationally representative estimates of health care use, expenditures, sources of payment, and insurance coverage for the U.S. civilian non-institutionalized population.

MEPS consist of three surveys. The Household Component (HC) is the core survey and forms the basis for the Medical Provider Component (MPC) and part of the Insurance Component (IC). The MEPS household component (HC) survey collects data through contact with a household by a series of five rounds of interviews over a two and half year period. Data are collected each year on a

new sample of households to provide current estimates of health care expenditures.

The MEPS medical provider component (MPC) add-on certifies information on medical care events reported in the MEPS HC by contacting medical providers and pharmacies identified by household respondents. The MPC sample includes all hospitals, hospital physicians, home health agencies, and pharmacies reported in the MEPS-HC. The MPC is conducted through telephone interviews and mailed survey materials. Sometimes, providers sent medical and billing records that were abstracted into the survey instruments.

The MEPS insurance component (IC) collects data on health insurance plans obtained through employers, unions, and other sources of private health insurance. Data obtained in the IC include the number and types of private insurance plans offered, benefits associated with these plans, premiums, contributions by employers and employees, eligibility requirements, and employer characteristics.

Together, these surveys yield comprehensive data that provide national estimates of the level and distribution of health care use and expenditures, support health services research, and can be used to assess health care policy implications.

The data set we used consisted of the Medical Expenditure Panel Survey (MEPS) information on prescribed antibiotics, using data from 1996-2004, with 2004 the most current year posted. The MEPS prescription antibiotics datasets consist of the following datasets for the years 1996-2004:

HC-010A	1996
HC-016A	1997
HC-026A	1998
HC-033A	1999
HC-051A	2000
HC-059A	2001
HC-067A	2002
HC-077A	2003
HC-085A	2004

These datasets were in SAS transport format, so the following code was written to import the data directly into SAS:

```
Libname Tseries V8 'F:\Dissertation';

FILENAME IN1 'F:\Disertation\H51A.SSP';
PROC XCOPY IN=IN1 OUT=TSERIES IMPORT;
RUN;
```

SAS CODE 1

Table 2.1 SAS code used to transport dataset in SAS format

For each data set, we created a STARTMEDDATE (start date of antibiotic), and the number of prescriptions for each antibiotic (DRUG). The following SAS code was written:

```
DATA SASUSER.H33A1;
SET SASUSER.H33A;
KEEP DUID RXBEGDD RXBEGMM RXBEGYR STARTMEDDATE RXICD1X
PURCHRD RXNAME RXQUANTY
RXMRX RXMDX RXPVX RXXPX DRUG;
IF RXBEGDD THEN DO;
RXBEGDD=ABS (RXBEGDD);
STARTMEDDATE=MDY (RXBEGMM, RXBEGDD, RXBEGYR);
END;
IF RXQUANTY >0 THEN DO;
DRUG=1;
END; RUN;
```

SAS CODE 2

Table 2.2 SAS code used to create the format of the dataset

Table 2.3 gives the description of the variables for the dataset, antibiotics. We have mentioned earlier that the variable, STARTMEDDATE, was formed by concatenating RXBEGDD (day antibiotic taken), RXBEGMM (month antibiotic taken), and RXBEGYR (year antibiotic taken).

Variable	DESCRIPTION
DUID	Unique Person Identifier
RXQUANTY	Quantity of prescribed antibiotic
RXMRX	Amount paid by Medicare
RXMDX	Amount paid by Medicaid
RXPVX	Amount paid by private insurance
RXXPX	Total amount paid (Cost of antibiotic)
STARTMEDDATE	Start date of antibiotic
RXICD1X	ICD9-CODE

Table 2.3. The variables and their descriptions

We created a time series variable by merging the day, month and year of each prescription with the label, STARTMEDDATE. If one of these three values is missing, there will be a missing value for STARTMEDDATE. A missing day is replaced by the beginning of the month and a missing month and year were deleted from the dataset; otherwise, imputing a value might distort the time series of the data.

We then merged each year's data set to create an antibiotics dataset for the years 1996-2004 combined. The following SAS code was written to do the task.

```
data Sasuser.Antibiotics;
length DUID 8 PURCHRD 8 RXBEGDD 8 RXBEGMM 8 RXBEGYR 8
RXNAME $ 50 RXQUANTY 8 RXICD1X $ 3 RXMRX 8 RXMDX 8 RXPVX8
RXXPX 8 STARTMEDDATE 8  DRUG 8;
Set Tseries.H10a Tseries.H16a Tseries.h26a Tseries.h33a Tseries.h51a
Tseries.h59a  Tseries.h67a  Tseries.h77a Tseries.h85a Tseries.hc10a ;
keep DUID PURCHRD RXBEGDD RXBEGMM RXBEGYR RXNAME
RXQUANTY RXICD1X RXMRX RXMDX RXPVX RXXPX
STARTMEDDATE DRUG;
run;
```

SAS CODE 3

Table 2.4 SAS code used to merge the dataset for each year

When the variable to be analyzed is text, each text is considered as a single variable level. For instance, Keflex 500MG and Keflex 250MG will be considered two different antibiotics if the text is not changed to Keflex. In order to avoid this problem for each antibiotic investigated in this project, we did filter to a commonly known antibiotic. The following code was written:

```
PROC SQL; CREATE TABLE SASUSER.ANTIBIOTICSFiltered AS SELECT

Sasuser.DUID FORMAT=BEST12, Sasuser.PURCHRD FORMAT=BEST12.,

Sasuser.RXBEGDD FORMAT=BEST12.,

Sasuser.RXBEGMM FORMAT=BEST12.,

Sasuser.RXBEGYR FORMAT=BEST12.,

Sasuser.RXNAME FORMAT=$F50., Sasuser.RXQUANTY FORMAT=BEST12.,

Sasuser.RXICD1X FORMAT=$F3.,

Sasuser.RXMRX FORMAT=BEST12., Sasuser.RXMDX FORMAT=BEST12.,

Sasuser.RXPVX FORMAT=BEST12., asuser.RXXPX FORMAT=BEST12.,

Sasuser.STARTMEDDATE FORMAT=DATE9.,

Sasuser.DRUG FORMAT=BEST12.,

((CASE WHEN "KEFLEX 250MG"=Sasuser.RXNAME THEN "KEFLEX"

WHEN "KEFLEX  500MG"= Sasuser.RXNAME THEN "KEFLEX"

WHEN "KEFLEX (CEPHELEXIN)"= Sasuser.RXNAME THEN "KEFLEX"
WHEN "KEFLEX - GENERIC"= Sasuser.RXNAME THEN "KEFLEX"

                datalines deleted …

WHEN "LEVAQUIN (FILM-COATED)"= Sasuser.RXNAME THEN "LEVAQUIN"
WHEN "LEVAQUIN 500MG"= Sasuser.RXNAME THEN "LEVAQUIN"

WHEN "LEVAQUIN 500MG TABS*50"= Sasuser.RXNAME THEN "LEVAQUIN"

WHEN "LEVAQUIN LEVA-PAK (3X5,FILM-COATED)"= Sasuser.RXNAME
THEN "LEVAQUIN"

ELSE Sasuser.RXNAME END )) AS Recode_RXNAME

FROM SASUSER.Antibiotics AS Antibiotics; QUIT;
```

SAS CODE 4

Table 2.5 SAS code used to match observations

Once the dataset was processed and filtered, we selected a set of antibiotics. An antibiotic is a drug that kills or prevents the growth of bacteria but has no effect against viruses or fungal infections. Antibiotics are less harmful to the patient or host compared to the infection, and therefore, can be used to treat infection.

When analyzing time series data, each observation should have a unique identifier such that this identifier is the time upon which the event happened. Due to scenarios that a patient might receive prescription antibiotics multiple numbers of times during the time interval under study, there might be several transactions made at a unique time point. Our data set has multiple prescriptions at the same time point, and time series models don't analyze such transactional data. Therefore, we converted the time identifier by accumulating the observations monthly, with the seasonal period set at 12 months.

The time period used to accumulate data points depends on what the forecaster wants to know. For instance, if we want to know the average number of prescriptions made during a month, then the accumulation will be an average. Since not every month has an equal numbers of days, we divide the total prescriptions made by the number of days in that particular month. On the other hand, if the forecaster wants to know the total number of prescriptions made during a particular month, then the accumulation point will be a sum. The accumulation point also could be the median, standard deviation, and so forth.

Accumulating data points depends on the specific research question needed to be addressed.

	RXBEGDD	RXBEGMM	RXBEGYR	RXNAME	RXQUANTY	RXICD1X	RXMRX	RXMDX	RXPVX	RXXPX	STARTMEDDATE
1121	8	9	1996	AMOXICILLIN	150	473	0	0	4.12	9.12	08SEP1996
1122	8	9	1996	AMOXICILLIN	90	490	0	7.21	0	7.21	08SEP1996
1123	8	9	1996	AMOXICILLIN	30	473	0	0	0	24.79	08SEP1996
1124	8	9	1996	AMOXICILLIN	150	382	0	0	0	8.99	08SEP1996
1125	8	9	1996	AMOXICILLIN	150	112	0	0	4.63	6.63	08SEP1996
1126	8	9	1996	AMOXICILLIN	21	473	0	0	0.3	5.3	08SEP1996
1127	9	9	1996	AMOXICILLIN	150	034	0	0	0	5.43	09SEP1996
1128	9	9	1996	AMOXICILLIN	24	008	0	8.49	0	8.99	09SEP1996
1129	9	9	1996	AMOXICILLIN	30	462	0	0	0	7.29	09SEP1996
1130	9	9	1996	AMOXICILLIN	30	460	0	0	3.43	8.43	09SEP1996
1131	9	9	1996	AMOXICILLIN	150	460	0	7.7	0	7.7	09SEP1996
1132	10	9	1996	AMOXICILLIN	30	382	0	0	0	5.62	10SEP1996
1133	10	9	1996	AMOXICILLIN	150	478	0	0	0	6.36	10SEP1996
1134	11	9	1996	AMOXICILLIN	30	008	0	0	6.48	10.48	11SEP1996
1135	11	9	1996	AMOXICILLIN	42	473	0	0	3.39	8.39	11SEP1996
1136	11	9	1996	AMOXICILLIN	30	379	0	0	0	10.44	11SEP1996
1137	11	9	1996	AMOXICILLIN	150	034	0	0	1.41	5.41	11SEP1996
1138	11	9	1996	AMOXICILLIN	150	034	0	0	7.22	7.22	11SEP1996
1139	11	9	1996	AMOXICILLIN	33	008	0	0	0	11.01	11SEP1996
1140	11	9	1996	AMOXICILLIN	150	382	0	0	3.7	6.14	11SEP1996
1141	12	9	1996	AMOXICILLIN	30	008	0	0	8.98	11.98	12SEP1996
1142	12	9	1996	AMOXICILLIN	150	382	0	0	0.49	5.49	12SEP1996
1143	12	9	1996	AMOXICILLIN	21	477	0	0	3.18	9.99	12SEP1996
1144	12	9	1996	AMOXICILLIN	150	382	0	6.69	0	6.69	12SEP1996
1145	13	9	1996	AMOXICILLIN	30	478	0	0	0	10	13SEP1996
1146	13	9	1996	AMOXICILLIN	21	474	0	0	0	13	13SEP1996
1147	13	9	1996	AMOXICILLIN	21	460	0	8.07	0	9.07	13SEP1996
1148	13	9	1996	AMOXICILLIN	45	518	0	0	2.23	10.79	13SEP1996
1149	13	9	1996	AMOXICILLIN	150	382	0	0	6.99	8.99	13SEP1996
1150	13	9	1996	AMOXICILLIN	150	382	0	0	6.99	8.99	13SEP1996
1151	13	9	1996	AMOXICILLIN	150	382	0	0	6.99	8.99	13SEP1996

Figure 2.1. The Prescription Data of Antibiotics.

Figure 2.1 represents the format of the antibiotics data set analyzed in this project. We can see that there are four transactions made on September 12, 1996. This is an indication of the data set as a transactional series. Time series only analyze equal time intervals; as a result, we accumulated the data. As the main aim of this study is to analyze prescription practices of antibiotics monthly, we forced the accumulation point to be total (Figure 2.2). It is of interest to know the average number of prescriptions of antibiotics, so we accumulated on average (Figure 2.3). When data are accumulated monthly, the total amount in that month is divided by the number of days in that month. SAS Code 5 and SAS

Code 6 create a time series for antibiotic data by accumulating total and average respectively.

```
proc hpf data=data.amoxicillin out=sasuser.amoxicilintotal lead=0;
        id startmeddate interval=month accumulate=TOTAL;
 forecast  RXQUANTY RXMRX RXMDX RXPVX RXXPX;
 by Rxname;
run;
```

SAS CODE 5

Table 2.6 SAS code to create dataset Amoxicillin accumulation of monthly prescriptions

VIEWTABLE: Work.Amoxicilintotal

	RXNAME	STARTMEDDATE	RXQUANTY	RXMRX	RXMDX	RXPVX	RXXPX
1	AMOXICILLIN	JAN1996	24705	0	390.34	847.75	3035.07
2	AMOXICILLIN	FEB1996	11591	0	166.82	316.54	1591.82
3	AMOXICILLIN	MAR1996	14155	2.45	284.11	456.83	1904.18
4	AMOXICILLIN	APR1996	7739	0	240.59	267.97	1237.52
5	AMOXICILLIN	MAY1996	6008	0	58.44	185.81	695.45
6	AMOXICILLIN	JUN1996	4681	0	100.21	97.09	452.31
7	AMOXICILLIN	JUL1996	3603	0	42.16	165.33	611
8	AMOXICILLIN	AUG1996	5425	0	65.3	154.25	603.66
9	AMOXICILLIN	SEP1996	7533	3.35	171.27	241.55	976.1
10	AMOXICILLIN	OCT1996	4412	0	85.47	185.56	727.79
11	AMOXICILLIN	NOV1996	4716	0	95.72	133.6	549.31
12	AMOXICILLIN	DEC1996	5862	0	111.58	148.78	600.95
13	AMOXICILLIN	JAN1997	14361	0	391.3	385.77	2143.77
14	AMOXICILLIN	FEB1997	13796	4.34	398.28	241.6	1774.68
15	AMOXICILLIN	MAR1997	13418	0	323.4	214.93	1693.57
16	AMOXICILLIN	APR1997	6499	40.78	371.91	195.24	1155.14
17	AMOXICILLIN	MAY1997	4492	10.48	108.15	147.94	600.33
18	AMOXICILLIN	JUN1997	3606	2.92	134.02	87.36	591.99
19	AMOXICILLIN	JUL1997	3714	0	120.97	93.15	467.59
20	AMOXICILLIN	AUG1997	3804	0.7	126.86	139.74	620.88
21	AMOXICILLIN	SEP1997	7355	5.97	296.11	168.89	971.23
22	AMOXICILLIN	OCT1997	7371	0	157.87	196.61	915.17
23	AMOXICILLIN	NOV1997	7942	0	184.33	164.07	1010.6
24	AMOXICILLIN	DEC1997	6387	9.72	89.4	277.68	1274.97
25	AMOXICILLIN	JAN1998	8314	0	99.72	193.89	954.16
26	AMOXICILLIN	FEB1998	9298	0	263.35	182.21	1116.2
27	AMOXICILLIN	MAR1998	7707	3.52	147.77	254.74	1023.14
28	AMOXICILLIN	APR1998	6251	0	73.84	142.82	712.57
29	AMOXICILLIN	MAY1998	6804	2.38	138.5	106.35	615.84
30	AMOXICILLIN	JUN1998	2785	5.02	59.21	116.71	381.45

Figure 2.2. The monthly prescriptions of Antibiotics

```
proc hpf data=data.amoxicillin out=sasuser.amoxicilinaverage lead=0;
        id startmeddate interval=month accumulate=Average;
forecast  RXQUANTY RXMRX RXMDX RXPVX RXXPX;
by Rxname;
run;
```

SAS CODE 6

Table 2.7 SAS code to create dataset Amoxicillin accumulation of average

VIEWTABLE: Work.Amoxicilinaverage

	RXNAME	STARTMEDDATE	RXQUANTY	RXMRX	RXMDX	RXPVX	RXXPX
1	AMOXICILLIN	JAN1996	74.189189189	0	1.1721921922	2.5457957958	9.1143243243
2	AMOXICILLIN	FEB1996	75.266233766	0	1.0832467532	2.0554545455	10.336493506
3	AMOXICILLIN	MAR1996	65.532407407	0.0113425926	1.3153240741	2.1149537037	8.8156481481
4	AMOXICILLIN	APR1996	57.753731343	0	1.7954477612	1.9997761194	9.2352238806
5	AMOXICILLIN	MAY1996	77.025641026	0	0.7492307692	2.3821794872	8.916025641
6	AMOXICILLIN	JUN1996	83.589285714	0	1.7894642857	1.73375	8.0769642857
7	AMOXICILLIN	JUL1996	61.06779661	0	0.7145762712	2.8022033898	10.355932203
8	AMOXICILLIN	AUG1996	73.310810811	0	0.8824324324	2.0844594595	8.1575675676
9	AMOXICILLIN	SEP1996	67.864864865	0.0301801802	1.542972973	2.1761261261	8.7936936937
10	AMOXICILLIN	OCT1996	57.298701299	0	1.11	2.4098701299	9.4518181818
11	AMOXICILLIN	NOV1996	81.310344828	0	1.6503448276	2.3034482759	9.470862069
12	AMOXICILLIN	DEC1996	81.416666667	0	1.5497222222	2.0663888889	8.3465277778
13	AMOXICILLIN	JAN1997	70.053658537	0	1.9087804878	1.881804878	10.457414634
14	AMOXICILLIN	FEB1997	68.297029703	0.0214851485	1.9716831683	1.196039604	8.7855445545
15	AMOXICILLIN	MAR1997	75.807909605	0	1.8271186441	1.2142937853	9.5681920904
16	AMOXICILLIN	APR1997	57.513274336	0.3608849558	3.2912389381	1.7277876106	10.222477876
17	AMOXICILLIN	MAY1997	72.451612903	0.1690322581	1.7443548387	2.3861290323	9.6827419355
18	AMOXICILLIN	JUN1997	61.118644068	0.0494915254	2.2715254237	1.4806779661	10.033728814
19	AMOXICILLIN	JUL1997	68.777777778	0	2.2401851852	1.725	8.6590740741
20	AMOXICILLIN	AUG1997	65.586206897	0.0120689655	2.1872413793	2.4093103448	10.704827586
21	AMOXICILLIN	SEP1997	68.101851852	0.0552777778	2.7417592593	1.5637962963	8.9928703704
22	AMOXICILLIN	OCT1997	67.009090909	0	1.4351818182	1.7873636364	8.3197272727
23	AMOXICILLIN	NOV1997	72.2	0	1.6757272727	1.4915454545	9.1872727273
24	AMOXICILLIN	DEC1997	51.096	0.07776	0.7152	2.22144	10.19976
25	AMOXICILLIN	JAN1998	72.295652174	0	0.8671304348	1.686	8.2970434783
26	AMOXICILLIN	FEB1998	71.523076923	0	2.0257692308	1.4016153846	8.5861538462
27	AMOXICILLIN	MAR1998	67.605263158	0.030877193	1.2962280702	2.2345614035	8.9749122807
28	AMOXICILLIN	APR1998	74.416666667	0	0.879047619	1.7002380952	8.4829761905
29	AMOXICILLIN	MAY1998	86.126582278	0.0301265823	1.753164557	1.3462025316	7.795443038
30	AMOXICILLIN	JUN1998	64.76744186	0.116744186	1.3769767442	2.7141860465	8.8709302326

Figure 2.3 Average prescriptions of Antibiotics

In this project, we will build a time series model to study the trend of prescription antibiotics. We selected a set of antibiotics to be analyzed based on the

availability of enough data points. This is due to the fact that some antibiotics are less prescribed compared to others and there are not enough data points collected to examine some of the sparsely used antibiotics. As a result, we selected a set of 20 antibiotics to investigate the trend of prescription practices of antibiotics. The description of the list of antibiotics investigated in this project is given in table 2.7.

Antibiotics	**Antibiotics**			
	Amoxicillin	Ampicillin	Azithromycin	Cefaclor
	Cefadroxil	Cefuroxime	Cephalexin	Cipro
	Clarithromycin	Clindamycin	Clotrimazole	Dicloxacillin
	Doxycycline	Erythromycin	Keflex	Sulfamethroxazole
	Tequin	Tetracycline	Tobramycin	Vancomycin

Table 2.8 Antibiotics Investigated for model forecast

Discussion

In this chapter, we have discussed how to prepare messy data for analysis. A statistical model analyzes a dataset to answer the research question of interest. In this project, we will build time series models to study and forecast prescription practices of antibiotics. Since we are building time series models, the dataset must have the form of time series data. A time series dataset is characterized by a series of fixed width time points. However, our dataset was transactional,

without any fixed time points. As a result, we preprocessed the dataset to have a format of time series. Once the messy dataset has time series variables, a model can be built to forecast and study the prescription practices of antibiotics.

CHAPTER 3

EXPONENTIAL SMOOTHING MODELS

Exponential smoothing is a class of techniques in time series analysis. Exponential smoothing produces good forecasts because it contains a self-adjusting mechanism for previous forecast errors, and because it assigns weights to previous data. In exponential smoothing, we assume that the future characteristics of the variable (Medicare, for example) are influenced by the past behavior of the variable. The main difference of exponential smoothing models from other time series models, for example ARIMA models, is that exponential smoothing assumes that a more recent event of a variable has more influence on future behavior compared to the distant past [12].

Exponential smoothing models are used as an extension of weighted averages. The mean of the observations is the sum of the numbers divided by the total number of observations. Mathematically,

$$\bar{Y} = \frac{1}{n}\sum_{t=1}^{n} Y_t = \frac{1}{n}(Y_1 + Y_2 + \Lambda + Y_n) = \frac{Y_1}{n} + \frac{Y_2}{n} + \Lambda + \frac{Y_n}{n} \qquad (3.1)$$

The weight for each variable Y_t in equation (3.1) is equal to $\frac{1}{n}$, where Y_t is the time series point at time t. Now, if we assume each Y_t is independent and identically distributed, then equation (3.1) is the best predictor of the next observed value, Y_{t+1}, using a criterion such as least squares or maximum likelihood. But in a time series, not all values have the same weight; recent values might have more weight compared to more distant values. Therefore, the general weighted average can be written as

$$\hat{Y} = \sum_{t=1}^{n} w_t Y_t, \ \sum_{t=1}^{n} w_t = 1. \qquad (3.2)$$

If the sample mean is used to forecast future values, the model used will be

$$Y_t = \mu + \varepsilon_t \qquad (3.3)$$

where $\varepsilon_t \sim N(0, \sigma^2)$, $E(\varepsilon_t, \varepsilon_j) = 0$, and $t \neq j$. If we assign a weight of one to the most recent observation, and a weight of zero to all other observations, we get a random walk model where the best predictor is the most recent observation, and all past observations provide no additional information to improve the prediction. Mathematically, the equation for a random walk model is

$$Y_t = Y_{t-1} + \varepsilon_t \qquad (3.4)$$

where $\varepsilon_t \sim N(0, \sigma^2)$, and if the random walk is moved by an average, the mathematical equation will be

$$Y_t = \mu + Y_{t-1} + \varepsilon_t \qquad (3.5)$$

Equation (3.5) is called a random walk with a drift.

If we compromise between the mean model and the random walk model, we get an exponential smoothing model where the weights exponentially decay to zero as we use older and older observations. To derive the equation for exponential smoothing models, we start with an infinite past and apply a weighted average with weights that exponentially decay to zero.

Given a time series Y_t for $t = 1,2,...,n$, the model assumed by the smoothing model has the following form:

$$Y_t = \mu_t + \beta_t t + s_p(t) + \varepsilon_t \qquad (3.6)$$

where μ_t represents the time varying mean term, β_t represents the time-varying slope, $s_p(t)$ represents the time-varying seasonal contribution for one of the p seasons and ε_t represent disturbances.

If the smoothing model of equation (3.6) doesn't have trend terms, then $\beta_t = 0$, and for a smoothing model without seasonal terms, $s_p(t) = 0$. If the smoothing model doesn't have time varying slope and time-varying seasonality, then equation (3.6) reduces to equation (3.3), which is the mean model.

The smoothing model estimates the time-varying components at each time t with the smoothing state. After initializing, the smoothing state is updated for each observation using the smoothing equations. The smoothing state at the last non-missing observation is used for prediction.

Smoothing State and Smoothing Equations

The smoothing equations determine how the smoothing state changes with increasing time. If we know the smoothing state at time t-1 and the time series value at time t, we can uniquely determine the smoothing state at time t. We have several smoothing models, but the smoothing state at time t consists of the following: L_t is a smoothed level that estimates μ_t, T_t is a smoothed trend that estimates β_t and $S_{t-j}, j = 0,1,..., p-1$ are seasonal factors that estimate $s_p(t)$.

Once we have the smoothing equation, we need to initialize the smoothing state. Let y_t be a time series where $t = 1,2,...,n$. Then the smoothing process first computes the smoothing state for time t=1. This computation requires an initial estimate of the smoothing state at time t=0, in which no data exist before time t=0. Therefore, we need to make an appropriate initial smoothing state where we back cast from the last time points, say t=n to t=1 to get a prediction at t=0. The smoothing state at time t=0 obtained from the back cast is used to initialize the smoothing process from time t=1 to t=n [10].

Seasonal Exponential Models

Exponential smoothing models give more weight to recent observations instead of to distant observations. Seasonal time series have strong autocorrelations that go far back into the past; therefore, simple exponential smoothing models don't capture seasonal effects. But if we add a seasonal component, then we can use the exponential smoothing model. The seasonal exponential smoothing method does not have a trend component.

An additive model is of the form $Y_t = \mu_t + s_p(t) + \varepsilon_t$, where μ_t is the linear term and $s_p(t)$ is the seasonal term. A multiplicative model (Winter's) is of the form $Y_t = (\mu_t + \beta_t t)s_p(t) + \varepsilon_t$, which is the product of the linear term and the seasonal

term. For the additive model, the seasonal factors sum to one, while for the multiplicative model, the seasonal factors average to one. For models with seasonal terms, we normalize the smoothing state so that $\sum_{j=0}^{p-1} S_{t-j} = 0$ for models that assume additive seasonality, and $\frac{1}{p}\sum_{j=0}^{p-1} S_{t-j} = 1$ for models that assume multiplicative seasonality such as Winter's method.

Statistically speaking, every model we build has an error term. The model that will be built for forecasting has to have a minimal error, i.e. the predictions should be made with minimal error. Predictions are made based on the last known value of the smoothing state.

Let $\hat{Y}_t(k)$ denote the prediction made at time t for k steps ahead, and let $e_t(k)$ denote the prediction error, where $e_t(k) = Y_{t+k} - \hat{Y}_t(k)$. As an example, a one step ahead prediction can be made at time $t-1$ for one time unit into the future, which can be denoted as $\hat{Y}_{t-1}(1)$. The one step-ahead prediction errors can be denoted by $e_t = e_{t-1}(1) = Y_t - \hat{Y}_{t-1}(1)$. The data set analyzed for this project has some missing values at certain time points. When a missing value is obtained by chance at time t, the smoothed values are updated using the error-correction form of the smoothing equations with the one step-ahead prediction error, e_t, set to zero. The missing value is estimated using the one-step-ahead prediction at time t-1, i.e. $\hat{Y}_{t-1}(1)$[11].

Exponential smoothing models are categorized into four basic models. They are the simple exponential smoothing model, the double exponential smoothing model, the linear exponential smoothing model and the damped trend exponential smoothing model.

Simple Exponential Smoothing Model

Given a time series Y_t, where $t = 1,2,...,n$, the model equation for simple exponential smoothing is given by

$$Y_t = \mu_t + \varepsilon_t \tag{3.7}$$

The smoothing equation is given by

$$\hat{Y}_{t+1} = \omega Y_t + \omega(1-\omega)Y_{t-1} + \omega(1-\omega)^2 Y_{t-2} + \omega(1-\omega)^3 Y_{t-3} + ... \tag{3.8}$$

If we factor out $(1-\omega)$ from equation (3.8), then we will have

$$\hat{Y}_{t+1} = \omega Y_t + (1-\omega)[\omega Y_{t-1} + \omega(1-\omega)Y_{t-2} + \omega(1-\omega)^2 Y_{t-3} + ...];$$

hence, equation (3.8) can be reduced to

$$\hat{Y}_{t+1} = \omega Y_t + (1-\omega)\hat{Y}_t \qquad (3.9)$$

where ω is the exponential weight.

Double (Brown) Exponential Smoothing Model

A double exponential smoothing model is applied to the data if a trend or seasonal factors appear in the data. A trend can be either increasing or decreasing. Double exponential smoothing considers a sequence of regression coefficients weighted more heavily towards the recent past.

The model equation for double exponential smoothing is

$$Y_t = \beta_0 + \beta_1 t + \varepsilon_t \qquad (3.10)$$

where β_0 and β_1 vary with time. The prediction for k units in the future from time t can be expressed in terms of a seasonal factor S_t and trend term T_t. Mathematically, the seasonal term can be expressed as $S_t = \omega S_t + (1-\omega)S_{t-1}$ and the trend term is expressed as $T_t = \omega S_t + (1-\omega)T_{t-1}$; hence, the double exponential smoothing model can be expressed as:

$$\hat{Y}_{t+k} = \left(2 + \frac{\omega^k}{1-\omega}\right) S_t - \left(1 + \frac{\omega^k}{1-\omega}\right) T_t \qquad (3.11)$$

The estimate of the slope at time t is given by $\hat{\beta}_{1t} = \frac{\omega}{1-\omega}(S_t - T_t)$ and an estimate of intercept at time t is given by $\hat{\beta}_{0t} = 2S_t - T_t - t\hat{\beta}_{1t}$. When we have real data with a trend and seasonality, the double exponential smoothing model will be applied where starting values for the slope and intercept are initialized. The starting values for the slope and intercept are $S_0 = \hat{\beta}_{0,0} - \left(\frac{1-\omega}{\omega}\right)\hat{\beta}_{1,0}$ and $T_0 = \hat{\beta}_{0,0} - 2\left(\frac{1-\omega}{\omega}\right)\hat{\beta}_{1,0}$ respectively.

Linear (Holt) Exponential Smoothing Model

The linear exponential smoothing model employs two time varying parameters while double exponential smoothing employs a time varying linear regression model that only relies on one smoothing parameter. As with the double exponential smoothing model, a linear exponential smoothing model is applied to model data with trend and seasonality.

For the linear (Holt) exponential smoothing, we begin with the same linear model as double exponential smoothing. However, a second parameter γ (trend smoothing weight) is added to equation (3.11). The model equation for double exponential smoothing is given by:

$$\hat{Y}_{t+k} = S_t + kT_t \qquad (3.12)$$

where $S_t = \omega Y_t + (1-\omega)(S_{t-1} + T_{t-1})$ and $T_t = \gamma(S_t - S_{t-1}) + (1-\gamma)T_{t-1}$.

Damped Trend Linear Exponential Smoothing Model

A damped trend linear exponential smoothing model is a modification to the linear (Holt) exponential smoothing model where the contribution of past trend is dampened or reduced. Adding a third parameter ϕ (damping coefficient) to equation (3.12) gives damped trend linear exponential smoothing. The model equation for damped trend linear exponential smoothing is given by:

$$\hat{Y}_{t+k} = S_t + \left[\sum_{i=1}^{k} \phi^i\right] T_t \qquad (3.13)$$

where $S_t = \omega Y_t + (1-\omega)(S_{t-1} + \phi T_{t-1})$ and $T_t = \gamma(S_t - S_{t-1}) + (1-\gamma)\phi T_{t-1}$.

Discussion

Exponential smoothing is one of several techniques in time series forecasting models. It is characterized by giving heavier weights to recent events and lower weights to past events. To reduce forecasting error, an optimal smoothing constant (ω) needs to be chosen. The ideal situation is to vary the smoothing constant (ω) until the prediction error is minimal. The main point we have to keep in mind is that regardless of which exponential model is chosen (simple, double, linear and damped trend), the exponential model we build has to minimize the residual (actual value-forecasted value).

To demonstrate the use of exponential smoothing models, we will build model forecasts for the antibiotics dataset. According the nature of the time series, one of the exponential smoothing models will be built to forecast antibiotic prescriptions. Once the exponential smoothing model is built for the antibiotic dataset, the forecasted values will be clustered into several clusters and compared to each other based on the severity of the disease.

Chapter 4

ARIMA MODELS

A time series that is a linear function of p past values plus an error is called an autoregressive process of order p, denoted by $AR(p)$.

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \varepsilon_t \quad (4.1)$$

where $\phi_1, \phi_2, ..., \phi_p$ are constants and ε_t is a white noise series with mean zero and variance σ_ε^2.

A time series that is a linear function of q past errors is called a moving average process of order q, denoted by $MA(q)$.

$$Y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_q \varepsilon_{t-q} \quad (4.2)$$

where $\theta_1, \theta_2, ..., \theta_q$ are parameters that determine the overall pattern of the process and ε_t is a white noise. A time series that is a linear function of p past

values plus a linear combination of q past errors is called an autoregressive moving average (ARMA) process of order (p,q), denoted by $ARMA(p,q)$:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_q \varepsilon_{t-q} \quad (4.3)$$

Sometimes an ARMA model is called a mixture model, i.e. a mixture of autoregressive and moving average parts. Another thing to be checked when building an ARMA model is to see if the series is stationary. A time series is stationary if the mean and variance of the series is constant at all time points.

Box-Jenkins (1976) modeling (also called ARIMA modeling) removes the non-stationary components such as trend and seasonality, and modeling the stationary part that is left over. Therefore, if a time series has a nonstationarity component, then an ARIMA model is used to forecast the series. So, in order to check whether the stochastic process is stationary, the following conditions must be satisfied for all values of t:

$$E(Y_t) = \mu \quad (4.4)$$

$$E[(Y_t - \mu)^2] = \sigma_y^2 = \gamma(0) \quad (4.5)$$

and

$$E[(Y_t - \mu)(Y_{t-\tau} - \mu)] = \gamma(\tau), \ \ \tau = 1,2,... \qquad (4.6)$$

Equations (4.4) and (4.5) define the mean and variance of the time series data, while equation (4.6) gives the autocovariance at lag τ. A covariance stationary stochastic process is a sequence of uncorrelated random variables with constant mean and variance; a process of this kind is known as a *white noise process* that is denoted by ε_t, which is assumed normally distributed with mean 0 and a variance of σ^2.

The variables in a white noise sequence are uncorrelated; hence the autocovariances at non-zero lags are all zero. Thus,

$$E(\varepsilon_t \varepsilon_{t-\tau}) = \begin{cases} \sigma^2, & \tau = 0 \\ 0, & \tau \neq 0 \end{cases}$$

The time domain properties of a stationary stochastic process can be summarized by plotting $\tau(t)$ vs τ, which is known as the *autocovariance function.* Standardizing autocovariances, that is, dividing through by the variance process gives the autocorrelation function, which is written as:

$$\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)}, \ \ \tau = 0,1,2,... \qquad (4.7)$$

The back shift of a series denoted by B shifts a time series by one time unit. The first lag is given by,

$$B(Y_t) = Y_{t-1}. \tag{4.8}$$

Applying B to equation (4.8) yields $B^2(Y_t) = B(B(Y_t)) = B(Y_{t-1}) = Y_{t-2}$, so by repeating this process, we find

$$B^{\tau}(Y_t) = Y_{t-\tau} \text{ for } \tau = 1,2,3,... \tag{4.9}$$

Autoregressive Process

An autoregressive process regresses a time variable on its own past values. An autoregressive process of order p is written as

$$Y_t = \phi_1 Y_{t-1} + ... + \phi_p Y_{t-p} + \varepsilon_t,\ t = 1,2,...,T, \tag{4.10}$$

so we can denote $Y_t \sim AR(p)$. The back shift operator lags an observation back into the past; for example, $B(Y_t) = Y_{t-1}$. As a result, we can write equation (4.10) using the back shift operator as

$$(1-\phi_1 B-\phi_2 B^2-...-\phi_p B^p)Y_t=\varepsilon_t \qquad (4.11)$$

Letting $\phi(B)=1-\phi_1 B-\phi_2 B^2-...-\phi_p B^p$, we can write equation (4.11) as

$$\phi(B)Y_t=\varepsilon_t \qquad (4.12)$$

The simplest autoregressive (AR) model is the first order model, denoted by AR(1). The first-order Autoregressive model is denoted by:

$$Y_t=\phi Y_{t-1}+\varepsilon_t, \qquad t=1,...,T. \qquad (4.13)$$

Using algebraic manipulation, (4.13) can be written as

$$Y_t-\phi Y_{t-1}=\varepsilon_t \Leftrightarrow (1-\phi_1 B)Y_t=\varepsilon_t. \qquad (4.14)$$

Here, we have used the polynomial operating on the time series to produce a white noise error sequence for the AR(1) model.

The series is assumed to have started at time $t=1$, but the process is regarded to have started in the remote past. Substituting repeatedly for lagged values of Y_t gives

$$Y_t = \sum_{j=0}^{J} \phi^j (\varepsilon_{t-j} + Y_{t-j}) = \sum_{j=0}^{J-1} \phi^j \varepsilon_{t-j} + \phi^J Y_{t-J} \qquad (4.15)$$

Taking the expectations of equation (4.15) and treating Y_{t-j} as a fixed number, we obtain

$$E(Y_t) = E(\sum_{j=0}^{J-1} \phi^j \varepsilon_{t-j}) + E(\phi^J Y_{t-J}) = 0 + \phi^J Y_{t-J} = \phi^J Y_{t-J} \qquad (4.16)$$

since $E(\sum_{j=0}^{J-1} \phi^j \varepsilon_{t-j}) = \sum_{j=0}^{J-1} \phi^j E(\varepsilon_{t-j}) = \sum_{j=0}^{J-1} \phi^j \times 0 = 0$. As mentioned in the beginning of this chapter, a time series is stationary if the mean and variance of the series at any time $t = j$ is constant.

We have shown that equation (4.14) is a representation of a polynomial operator on the time series to produce a white noise error sequence; in fact, equation (4.14) is a representation of an AR(1) model. We can also represent an AR(p) process by a characteristic polynomial, $\phi(x) = 1 - \phi_1 x - \phi_2 x^2 - ... - \phi_p x^p$.

Another way of checking if a series is stationary is to see if the roots of the characteristic equation are less than one in absolute value. For an autoregressive function of order p, i.e. AR (*p*), the characteristic equation is given by

$$x^p - \phi_1 x^{p-1} - ... - \phi_p = 0 \qquad (4.17)$$

Replacing x by $1/B$ in equation (4.16) and multiplying by B^p, we obtain a polynomial equation:

$$1 - \phi_1 B - ... - \phi_p B^p = 0 \qquad (4.18)$$

In order for AR(p) to be stationary, the roots of equation (4.17) should be less than one in magnitude or the root of equation (4.18) should be greater than one in magnitude.

Moving Average Processes

A moving average process of order q is written as

$$Y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_q \varepsilon_{t-q},\ t = 1,2,...,T \text{ and } \varepsilon_t \sim N(0, \sigma^2) \qquad (4.19)$$

and is denoted by $Y_t \sim MA(q)$.

Let $\{Y_t\}$ be the MA(q) model given in equation (4.19). Then the following hold:

$$E(Y_t) = 0 \tag{4.20}$$

$$\text{var}(Y_t) = (1 + \theta_1^2 + ... + \theta_q^2)\sigma^2 \tag{4.21}$$

$$\text{cov}(Y_t, Y_{t+k}) = \begin{cases} 0, & |k| > q, \\ \sigma^2 \sum_{i=0}^{q-|k|} \theta_i \theta_{i+|k|}, & |k| \leq q, \end{cases} \tag{4.22}$$

Equation (4.22) can be shown as follows

$$\text{cov}(Y_t, Y_{t+k}) = E(Y_t Y_{t+k})$$

$$= E(\varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + ... + \theta_q\varepsilon_{t-q})(\varepsilon_{t+k} + \theta_1\varepsilon_{t+k-1} + \theta_2\varepsilon_{t+k-2} + ... + \theta_q\varepsilon_{t+k-q})$$

$$= \sigma^2 \sum_{i=0}^{q-|k|} \theta_i \theta_{i+|k|} \text{ , where } \theta_0 = 1$$

The correlation at lag k is given by the following equation

$$\rho(k) = \begin{cases} \sum_{i=1}^{q-|k|} \theta_i \theta_{i+|k|} / \sum_{i=0}^{q} \theta_i^2, & |k| \leq q, k \neq 0 \\ 1, & k = 0 \\ 0 & \textit{otherwise.} \end{cases} \tag{4.23}$$

The simplest moving average (MA) model is the first order model, denoted by MA(1). The first-order moving average model is denoted by:

$$Y_t = \varepsilon_t + \theta\varepsilon_{t-1}, \qquad t = 1,...,T \quad \varepsilon_t \sim N(0,\sigma^2) \qquad (4.24)$$

Substituting repeatedly for lagged values of Y_t, equation (4.24) can be written as

$$Y_t = \theta Y_{t-1} - \theta^2 Y_{t-2} + \Lambda \; -(-\theta)^J Y_{t-J} + \varepsilon_t - (-\theta)^{J+1}\varepsilon_{t-J-1} \qquad (4.25)$$

$$= -\sum_{i=1}^{J-1}(-\theta)^i Y_{t-i} + \varepsilon_t$$

We have seen that in order for AR (p) to be stationary, the roots of the characteristic polynomial of equation (4.18) must be larger than one in magnitude, whereas for MA (q), we have to consider invertability since the autocorrelation function after lag *q* vanishes for MA(q); it is stationary.

Let $\{Y_t\} \sim MA(q)$, where Y_t is given as equation (4.19). Then we can write equation (4.19) as $Y_t = \theta(B)\varepsilon_t$, where $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + ... + \theta_q B^q$ with $B(\varepsilon_t) = \varepsilon_{t-1}$. Therefore, in order for MA(q) to be invertible, the roots of the equation $\theta(B) = 0$ must all lie outside the unit circle[13].

ARMA MODELS

The AR (p) and MA (q) models are combined to give an ARMA (p, q). By combining these models, we get an ARMA(p,q) model with few unknown parameters. It should also be noted that both AR (p) and MA (q) are special cases, so it is legitimate to denote them by $ARMA(p,0)$ and $ARMA(0,q)$ respectively. An ARMA Model is represented as:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \Lambda + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \Lambda + \theta_q \varepsilon_{t-q}$$

(4.26)

A time series $\{Y_t\}$ is said to be an ARMA(p,q) process if

(i) $\{Y_t\}$ is stationary

(ii) $\forall t$, $\phi(B)Y_t = \theta(B)\varepsilon_t$ where $\varepsilon_t \sim N(0,\sigma^2)$.

White Noise

A process that is purely random is known as white noise. Let $\{\varepsilon_t\}_{-\infty}^{\infty}$ be a sequence of errors with mean and variance given by $E\{\varepsilon_t\} = 0$ and $E\{\varepsilon_t^2\} = \sigma^2$ respectively for which all the errors are uncorrelated across time, i.e. $E\{\varepsilon_t \varepsilon_\tau\} = 0$

for $t \neq \tau$. A process satisfying these three conditions is called a white noise process.

The characteristic polynomials are representations of Box-Jenkins models on which

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \Lambda - \theta_q B^q \text{ and } \phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \Lambda - \phi_p B^p .$$

The simplest autoregressive moving average model is the first order model, denoted by $ARMA(1,1)$. The first-order autoregressive moving average model is denoted by:

$$Y_t = \phi_1 Y_{t-1} + \varepsilon_t - \theta_1 \varepsilon_{t-1}, \qquad t = 1,...,T \quad \varepsilon_t \sim N(0, \sigma^2) \tag{4.27}$$

Alternatively, equation (4.27) can be written as $Y_t(1 - \phi_1 B) = \varepsilon_t(1 - \theta_1 B)$. Then

$$Y_t - \phi_1 Y_{t-1} = \varepsilon_t - \theta_1 \varepsilon_{t-1} \Leftrightarrow Y_t - \phi_1 B Y_t = \varepsilon_t - \theta_1 B \varepsilon_t \Leftrightarrow Y_t(1 - \phi_1 B) = \varepsilon_t(1 - \theta_1 B) .$$

This is the form of an $ARMA(1,1)$.

Non-Seasonal ARIMA Models

An ARIMA model is a generalization of an ARMA model on which differencing (d) is involved. Let $Z_t = (1-B)^d Y_t$ and assume that Z_t is an $ARMA(p,q)$, i.e. $\phi(B)Z_t = \theta(B)\varepsilon_t$. Then

$$\phi(B)(1-B)^d Y_t = \theta(B)\varepsilon_t . \qquad (4.28)$$

where

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \Lambda - \phi_p B^p$$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \Lambda - \theta_q B^q$$

The process $\{Y_t\}$ is said to be an $ARIMA(p,d,q)$, autoregressive integrated moving average model[14]. For instance an ARIMA(1,1,1) can be written as

$$(1-\phi B)(1-B)Y_t = \varepsilon_t - \theta\varepsilon_{t-1} . \qquad (4.29)$$

Since equation (4.29) has a difference of 1, then

$$Z_t = (1-B)Y_t = Y_t - BY_t = Y_t - Y_{t-1}.$$

Hence, summing $\sum_{i=1}^{t} Z_t = \sum_{i=1}^{t} (Y_k - Y_{k-1}) = Y_t - Y_0$ and assuming $Y_0 = 0$, Y_t can be derived from Z_t by summing, and that is why we have the term, "integrated" in an ARIMA model.

SEASONAL ARIMA MODELS

When a time series $\{Y_t\}$ exhibits a seasonal trend, then the ARIMA model will be used to fit the model with a seasonality term, and it is written as:

$$\phi(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D Y_t = \theta(B)\Theta_Q(B^s)\varepsilon_t \qquad (4.30)$$

where

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \Lambda - \phi_p B^p,$$

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \Lambda - \theta_q B^q,$$

$$\Phi_p(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{s^1} - \Lambda - \Phi_p B^{s^P},$$

$$\Theta_Q(B^s) = 1 - \Theta_1 B^s - \Theta_2 B^{s^1} - \Lambda - \Theta_Q B^{s^Q}.$$

Therefore, $\{Y_t\}$ is denoted by ARIMA$(p,d,q)\times(P,D,Q)_s$. When a series $\{Y_t\}$ has seasonality, then Y_t is related to a series that cycles using an s time unit, in the

past denoted by Y_s. To demonstrate, let us consider a seasonal ARIMA(1,0,0) $\times$ $(0,1,1)_{12}$ time series of $\{Y_t\}$. Then substituting $p=1, q=0, d=0, P=1, D=1, Q=12$ and $s=12$ in equation (4.30), we get

$$(1-\phi B)(1-B^{12})Y_t = \varepsilon_t - \theta\varepsilon_{t-12}$$

$$\Leftrightarrow (1-B^{12}-\phi B+\phi B^{13})Y_t = \varepsilon_t - \theta\varepsilon_{t-12}$$

$$\Leftrightarrow Y_t - B^{12}(Y_t) - \phi B(Y_t) + \phi B^{13}(Y_t) = \varepsilon_t - \theta\varepsilon_{t-12}$$

$$\Leftrightarrow Y_t - Y_{t-12} - \phi Y_{t-1} + \phi Y_{t-13} = \varepsilon_t - \theta\varepsilon_{t-12}$$

$$\Leftrightarrow Y_t = Y_{t-12} + \phi(Y_{t-1} - Y_{t-13}) + \varepsilon_t - \theta\varepsilon_{t-12}.$$

Here we see that Y_t depends upon $Y_{t-12}, Y_{t-1}, Y_{t-13}$ and ε_{t-12}.

Model Validation and Holdout Sample

Model validation is an important part of the model development process[28]. Once the model produced is validated, it can be used for decision making. For example, the forecasted model can be used as a model prediction for the cost of prescriptions. It is advisable to ask whether the model of prediction is validated or not. In practice, no model will fully fit the data perfectly (100%); there is some error to some extent. So in model building, our main task is to reduce the error as much as possible. Model validation assures that the prediction and forecast for the dataset analyzed is accurate and solves the problem being investigated.

Once the models built are validated, we can make future predictions or forecast for the prescription of antibiotics.

The forecasting models that we build for the prediction of the cost of antibiotics may work well with the data to which they were fit; we also have to consider how well the models will fit to freshly collected data. In order to solve this problem, we validate the model built. By model validation, we mean to test the dataset on another independent dataset and see how well the model will fit the new dataset. The model built should work on an independent dataset which was not used for fitting the model equations. For these reasons, we classify the dataset into portions; one of which is used for model fitting and the other for model validation. For this project, we will hold out 20% of the data for validation. By hold-out samples, we mean that a portion of the data is used for model evaluation and the rest of the data are used for model forecasting. If there is a hold-out sample, the period of fit is different from the period of evaluation, which we call a hold-out sample. The period of fit ends at a time point before the series of data ends, and the remainder of the data are held out as a non-overlapping period for evaluation. Hence, a hold-out sample is a period used to compare the forecasting accuracy of models fit to past data.

Model Diagnostics

In a time series model, each observation is correlated with past values, so there is some correlation of present values with past values. An autocorrelation function gives the primary diagnostics for a model fit. The autocorrelation function at lag k, denoted by ACF(k), represents the correlation of a time series with itself lagged by k time units. For a given model, we can check whether there is a need for differencing just by observing if the series autocorrelations at each lag die off quickly or not. If the autocorrelation dies quickly, there is no need for differencing.

The partial autocorrelation function, denoted by PACF(k) measures the correlation of a time series with itself at lag k adjusted for lags $1,2,...,k-1$. The partial autocorrelation function at lag k is the coefficient of the k^{th} order autoregressive term in an autoregressive model of order k. The partial autocorrelation function plot has similar diagnostic properties to the autocorrelation function plot.

The inverse autocorrelation function denoted by IACF(k) is an inverse of the autocorrelation function. The inverse autocorrelation function identifies model behavior not detected by the PACF. Chatfield (1980) suggests that IACF should replace the PACF as a model diagnostic tool[15].

When a series of data is non-stationary, we use an ARIMA model to produce a forecast. Nonstationarity is removed by taking differencing ($Y_t - Y_{t-1}$) for each series and then building an ARIMA model for the differenced data. If the differenced series of data is non-stationary, a second differencing ($Y_t - Y_{t-2}$) is taken for each series, building an ARIMA model for the differenced data. The antibiotics dataset has some missing values or zero for variables such as Medicare and Medicaid. We want to test whether this series is white noise residual, so it is crucial to do a white noise test.

The other model diagnostic statistics for an ARIMA model are Akaike's information criterion (AIC) (Akaike 1974; Harvey 1981)[16] and Schwarz's Bayesian criterion (SBC) (Schwarz 1978)[17]. Using AIC and SBC, we will compare several ARIMA models fit to the antibiotics dataset. We will choose a model with the smallest information criteria. Both AIC and SBC are called information criteria. The AIC is computed as

$$-2\ln(L) + 2k \tag{4.31}$$

where *L* is the likelihood function and *k* is the number of free parameters. The Schwarz's Bayesian criterion (SBC) is computed as

$$-2\ln(L) + \ln(n)k \tag{4.32}$$

where *n* is the number of residuals that can be computed for the time series. Sometimes, Schwarz's Bayesian criterion is called the Bayesian Information criterion (BIC).

A stationarity test is very important in time series modeling, especially when an ARIMA model is built. Hence, we will perform a stationarity test such as the Dickey-Fuller and Philips-Perron test.

Dickey-Fuller Test:

When a series has a unit root, the series is non-stationary and the ordinary least squares estimator is not normally distributed. The limiting distribution of the ordinary least squares estimator of autoregressive models for time series with a simple unit root was studied by Dickey and Fuller (1979)[18]. Dickey, Hasza, and Fuller (1984) obtained the limiting distribution for a time series with seasonal unit roots[19].

Let us consider the AR(p) model given in equation (4.10). If all of the characteristic roots of equation (4.17) are less than unity in absolute value, then $Y_t \sim AR(p)$ is stationary. If there is a unit root, then $Y_t \sim AR(p)$ is non-stationary

and the sum of the autoregressive parameters given by $\sum_{i=1}^{p}\phi_i$ in equation (4.10) is equal to one. As a result, we will test for a unit root by using the hypothesis H_0: $\sum_{i=1}^{p}\phi_i = 1$ against H_1: $\sum_{i=1}^{p}\phi_i \neq 1$. The Dickey-Fuller test is used to test the null hypothesis that the time series exhibits a lag *d* unit root against the alternative of stationarity. We will use the Dickey-Fuller test to test for stationarity and determine the order of differencing needed for the ARIMA modeling of the antibiotics time series data.

Phillips-Perron Test

The Phillips-Perron is another test statistic for testing stationarity or unit roots. It performs tests for zero mean and single mean in an autoregressive model. A zero mean for an autoregressive model is given by

$$Y_t = \phi Y_{t-1} + \varepsilon_t \qquad (4.33)$$

and a single mean is given by

$$Y_t = \mu + \phi Y_{t-1} + \varepsilon_t \qquad (4.34).$$

where ε_t ~ serially correlated.

We will compute the Phillips-Perron test for the null hypothesis that Y_t has a unit root against a stationary alternative. The Philips-Perron tests are similar to Dickey Fuller tests, but Philips-Perron tests add an automatic correlation to the Dickey Fuller test procedure to allow for autocorrelated residuals. For the Philips-Perron test, the errors are identically and independently distributed.

Discussion

Time series data are stationary if the mean and variance of the series is constant through time. However, in most situations, time series data are non-stationary. In order to build model prediction, we have to use an ARIMA model. An ARIMA model builds a model prediction for a non-stationary time series. An ARIMA model does differencing to the series of data, until the series of data becomes stationary; then a model for prediction is built. In most cases, we first do differencing, but if the differenced series remains non-stationary, we repeat the differencing until the series becomes stationary.

In statistics, we need to have a test to validate whether the selected model fits the data well or not. For the models we build using ARIMA models, we will check several model diagnostic statistics. As a model diagnostic, we will check Akaike's information criterion (AIC), Schwarz's Bayesian criterion, stationarity

tests such as Dickey Fuller and Philips-Perron. Once these test statistics are checked, we will build a model forecast for the antibiotics data set. We will also evaluate the model built by holding 20% of the dataset. The hold-out sample is used for model evaluation and the remaining 80% of the dataset will be used for model development.

Chapter 5

Heteroskedastic Models

There is an interest in forecasting not only the levels of the time series Y_t, but also its variance. The need to consider variance occurs when the series is more volatile at some times compared to others. When the prescription of antibiotic varies very often, the variance of the series is volatile, so we will be forecasting not only the level of the series Y_t but also the variance of the series. As a result, we will use the generalized autoregressive conditional heteroskedasticity (GARCH) model to forecast the prescription of antibiotics. The generalized autoregressive conditional heteroskedasticity (GARCH) model is an extension of the autoregressive conditional heteroskedasticity (ARCH) model. We will first describe the model equation for ARCH; from that, we will set up the model equation for GARCH [33].

In chapter four, we defined and explained the model equation for an autoregressive process of order p that is denoted by AR(p). The model equation for an AR(p) is given by

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \varepsilon_t . \qquad (5.1)$$

where ε_t is white noise. Given $Y_{t-1}, Y_{t-2}, ..., Y_{t-p}$, we forecast Y_t. An AR(p) is utilized to do this task by regressing the present value of Y_t on its past values.

In this project, we will consider the effect of the inflation rate on the cost of the prescription antibiotics. When there is a volatile inflation rate, it would be advisable to study changes in variance. As a result, the changing variance also has significance on the estimation of the parameters of equation (5.1) that describe the dynamic of the level of the time series variable Y_t.

In chapter four, we showed that the unconditional variance of the error term is constant; that means, $E(\varepsilon_t) = 0$, while $E(\varepsilon_\tau \varepsilon_t) = \sigma^2$ for $t = \tau$ and $E(\varepsilon_\tau \varepsilon_t) = 0$ for $t \neq \tau$. However, if we consider the conditional variance of the error term, ε_t may not be constant due to the volatility of variance. As a result, the error term could change from time to time. One remedy to control the volatility of an error term ε_t is to model the square of ε_t as an autoregressive order p denoted by AR(p). The modeling equation for the square of ε_t is given by:

$$\varepsilon_t^2 = \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \varepsilon_{t-2}^2 + \Lambda + \alpha_p \varepsilon_{t-p}^2 + \omega_t \qquad (5.2)$$

where ω_t is a white noise process for equation (5.2). The expected value of the white noise in equation (5.2) is zero such that $E(\omega_\tau \omega_t) = \beta^2$ for $t = \tau$ and $E(\omega_\tau \omega_t) = 0$ for $t \neq \tau$. We see that ε_t is the error in forecasting Y_t in equation (5.1). A white noise process ε_t satisfying equation (5.2) is known as an autoregressive conditional Heteroskedastic process of order p, denoted by $\varepsilon_t \sim ARCH(p)$. The conditional distribution of the square error term of a forecast of Y_t on the previous p squared forecast errors is given by:

$$\hat{E}(\varepsilon_t^2 \mid \varepsilon_{t-1}^2, \varepsilon_{t-2}^2, ...) = \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \varepsilon_{t-2}^2 + \Lambda + \alpha_p \varepsilon_{t-p}^2 \qquad (5.3)$$

For an ARCH model described in equation (5.2), we want to test whether the residuals ε_t from a regression model exhibit time-varying heteroskedasticity without estimating the parameter estimates such as $\alpha_1, \alpha_2, ..., \alpha_p$. The ARCH(p) process is regarded as an AR(p) process for the square of the error term in equation (5.1). Bollerslev (1986) recommended the use of an ARIMA model for analyzing the autocorrelations of ε_t^2 [29] as an alternative to ARCH(p).

We can also write an ARCH(p) process in a slightly different form than equation (5.2) as follows. Suppose that

$$\varepsilon_t = \sqrt{k_t}.v_t \qquad (5.4)$$

where $\{\varepsilon_t\}$ is an independent identically distributed (i.i.d.) sequence with zero mean and variance of one; mathematically, $E(v_t) = 0$ and $E(v_t^2) = 1$. If k_t is defined by

$$k_t = \alpha_1 \varepsilon_{t-1}^2 + \alpha_2 \varepsilon_{t-2}^2 + \Lambda + \alpha_p \varepsilon_{t-p}^2 \qquad (5.5)$$

and ε_t is generated by equation (5.4) and equation (5.5) then $\varepsilon_t \sim ARCH(p)$ process.

Generalized Autoregressive Conditional Heteroskedasticity Models

Engle (1982) introduced a model in which the variance at time t is modeled as a linear combination of past squared residuals, which is called an autoregressive, conditional, Heteroskedastic process (ARCH) [30]. On the other hand, if the variance model looks more like an ARMA than AR, we have a generalized autoregressive conditional Heteroskedastic model denoted by GARCH. Bollerslev (1986) introduced the GARCH models [29].

Although the requirement of equal means can be dealt with using differencing, there is no way of dealing with the requirement of equal variances, called

Homoskedasticity. On the other hand, if the variances are not equal, the series has heteroskedasticity. The GARCH(p,q) process models the residual of a time series regression. We now define a model for generalized ARCH as follows: let the time series Y_t be defined as

$$Y_t = X_t\beta + \varepsilon_t \qquad (5.6)$$

The residual is modeled as $\varepsilon_t = \sqrt{k_t}v_t$ where

$$k_t = \delta_1 k_{t-1} + \delta_2 k_{t-2} + \Lambda + \delta_p k_{t-p} + \theta_1 \varepsilon_{t-1}^2 + \theta_2 \varepsilon_{t-2}^2 + \Lambda + \theta_q \varepsilon_{t-q}^2 \qquad (5.7)$$

Equation (5.7) is the generalized autoregressive conditional heteroskedasticity model denoted by $\varepsilon_t \sim GARCH(p,q)$.

In the standard GARCH model, v_t has the unit Normal density, $\frac{1}{\sqrt{2\pi}} e^{\frac{-v_t^2}{2}}$. In addition, to preserve stationarity, the following constraints are placed on the coefficients: $k > 0$, $\delta \geq 0$, $\theta \geq 0$, and $\sum_i^p \theta_i + \sum_j^q \delta_j < 1$.

Discussion

The ARCH/GARCH models are used when the variance of the data is changing from time to time; that is when there is volatility. From the least squares model, we know that the expected value of the square of error terms is the same at any given point in time. In this case, we have Homoskedasticity. On the other hand, data on which the variances of the error terms are not equal to each other have heteroskedasticity when the error terms are higher at some time point and smaller at other time points. When there is heteroskedasticity in the error term, the regression coefficients for ordinary least squares regression are unbiased, but the standard errors and confidence intervals estimated will be too narrow, giving incorrect precision. The problem of heteroskedasticity is corrected using models ARCH and GARCH by modeling the variance [30].

A GARCH model is a weighted average of past squared residuals; it has a declining weight on which recent observations are given larger weight than distant, past observations. The GARCH models are good in predicting conditional variances, so for the prescription of antibiotics where there is volatile variability in the prescription of antibiotics, we will apply the GARCH model as a forecasting tool. We will also use the Lagrange multiplier to test to determine the order of the ARCH model appropriate for the data being analyzed. If the tests are significant and for which we have smaller p-values through a given order, very high-order ARCH model will be needed to model heteroskedasticity.

Chapter 6

Text Mining

Text mining is the discovery of new, previously unknown information by extracting information from different written sources[21]. Once the information is extracted, we link the information together to form new facts to be explored further. We can ask what makes text mining different from web mining.

Suppose we want to search for the word, antibiotics, in www.google.com. The information that would be displayed includes known facts about antibiotics, even though we might not be sure about the meaning of the term. Here, we are not finding new facts about antibiotics; the information was available from written sources. In text mining, we are looking for new information that was not known before. The new information that we found could be related to existing information. Text mining finds interesting patterns from large databases. For our database, we will use text mining to cluster the ICD-9 codes to a total of six clusters. Once we classify the ICD-9 codes into six clusters, the antibiotics can be compared to each other based on the probability density curve of the cost variables; for example, the private insurance payments made.

Definition of Text Mining

Text mining extracts the patterns from text language; on the other hand, data mining extracts patterns from structured databases. If we want to extract some information from a database using a data mining approach, a computer program is written to extract the needed information. In text mining, we don't write computer programs, but rather, we link the information needed from the text available. For instance, if we are looking for the word, diabetes, text mining analysis finds information that can be linked to diabetes, such as the term, insulin.

The Webster's online dictionary defines text mining as the process of extracting interesting and non-trivial information and knowledge from unstructured text[22]. In text mining, we structure the input text by parsing, adding and removing some derived linguistic features to derive patterns within the structured data so that output can be evaluated and interpreted. The ideal situation of using text mining is to classify and cluster the information so that a possible pattern will be discovered.

Another important question to be addressed is how we discover useful information from a large number of text documents. Nowadays, there has been a fast increase in the number of text sources on the internet, and the need to

extract important information from the text data has been growing. The question arises as to how we extract the information available in the web to get the information that we need. Text mining analyzes the unstructured web text data by finding information that is not immediately visible within the web documents using techniques from data mining, machine learning, and natural language processing [22].

It should be noted that text mining is not an information extraction methodology or a text summarization method. In order to classify a group of thousands of documents into identifiable categories without the need of reading every document, computer software is required.

Text mining is the automated or partially automated processing of text by imposing structure upon text and extracting useful information from text[24]. Clustering algorithms, such as feature map algorithms based on the self-organization map (SOM) [25], are frequently used for the purposes of text mining. However, algorithms such as SOM group the text into clusters based on similarity. Since we are using similarity, the textual data need to be converted to numbers, i.e. by converting the text into phrases or words and then encoding these phrases or words using techniques such as the Term Frequency-Inverse Document Frequency (TF-IDF) [26].

Text mining classifies the textual data into clusters so that meaningful information can be retrieved. From each text, words or phrases are collected and a classification is made based on the frequency of occurrence.

It should be noted that commonly existing words such as “a”, “is” and “of” are not used as part of the classification; as a result, they are put in the stop word list. A stop word list is a set of words not used in the classification[26]. For this project, we used a set of words such as ‘have’, ‘is’, ‘before” as a stop word list. As an application of text mining, let us consider physicians comments about patient conditions. There is a clear difference from physician to physician on the amount of text (words) written; some physicians are very brief, but others are wordy. As a result, there is a difference when analyzing, say some clinical condition, due to differences in the length of the wording.

In the previous paragraphs, we have talked about the use of text mining in clustering unstructured text. The antibiotics dataset we are investigating has a variable called ICD-9. ICD-9 is a categorical variable that contains a set of five digits that are used to classify each disease condition. The ICD-9 codes are used as textual content, so we have to devise a method to classify the ICD-9 codes into meaningful clusters. The antibiotics dataset has daily prescriptions for the years 1996-2004. Analyzing each ICD-9 code would be cumbersome; as a result, we will employ text mining to analyze and cluster the ICD-9 codes to a reasonable number of clusters. We will use a maximum of six clusters; once the

clusters are formed, we will use kernel density estimation to compare the probability distributions of each antibiotic.

The next question we discuss is how to analyze text data using SAS software. At this stage, we are considering for a given text datum how to classify it into an optimal cluster and analyze the text data results.

Document Frequency Matrix

Text mining creates frequency counts for each word or phrase classified in a cluster. It can be considered as statistical summaries where word frequency is the statistical summary of interest. When analyzing text data, each word or phrase is given a weight so that the document will be classified into groups. SAS Enterprise Miner uses Entropy, Global Freq/IDF (GF-IDF), Inverse Doc Freq (IDF), Normal, None, Chi-Squared, Mutual Information and Information Gain as term weightings. Raw counts don't provide much help in discriminating between documents; therefore, weighing schemes are derived to separate documents into groups or clusters.

In text mining, not all terms are important; some terms are more important compared to others. As a consequence, term weighting are used to help

designate the importance of the terms. Terms with the most discriminating power are those that occur most frequently, but in only a few documents.

Transforming the Term by Document Frequency Matrix

There are two main methods to work with a parsed matrix; they are singular value decomposition (SVD) and roll up terms. Text parsing is the preliminary step to text mining. Text parsing groups similar documents based on the terms used within the document, works with a frequency matrix table of terms by documents to cluster documents, and creates a term-document frequency matrix.

The singular value decomposition for a matrix B is defined as

$$B = U\Sigma V \qquad (6.1)$$

where U is the matrix of the term vector; Σ is a diagonal matrix with singular values along the diagonal; and V is the matrix of the document vector. Both U and V have orthonormal columns. The truncated decomposition of matrix B is when the SVD calculates only the first K columns of U, Σ and V. Each column or document in matrix B can be projected onto the first K columns of U, and each row or term in matrix B can be projected onto the first K columns of V. The

column projection of matrix B is a method used to represent each document by K different concepts. Therefore, for any collection of documents, the SVD forms a K-dimensional subspace that fits best to describe the data.

On the other hand, the method of Roll up Terms uses the highest term weights in the document collection. The term document frequency matrix is the number of roll up terms by the number of documents.

Analyzing the Text Data

Analysis of text data has three main purposes: exploratory analysis of the collection of documents (text data), clustering of data and finding relationships between terms. Statistical methods such as the nearest neighborhood algorithm use distance proximity to cluster observations. In text mining, we use hierarchical and expectation maximization to cluster the text data.

Expectation Maximization (EM) clustering considers the data as a combination of probability functions that are normally distributed. Suppose we want to cluster observations into two clusters, for two normally distributed populations with unknown parameters. Since the parameters of the two normal distributions are unknown, we use an iterative algorithm to estimate the parameters from the data. Once the parameter estimates are obtained, an observation is assigned to

cluster one if the density function for cluster one is larger than for cluster two at that observation; otherwise it will be assigned to cluster two.

We have seen that ICD-9 codes are a set of five digits which are used for a classification of a disease. ICD-9 codes are better if they are treated as text rather than as category because the similarity between the codes can be related to similarities in patient conditions, taking full advantage of the stemming properties within the codes [27]. In this project, we will use text mining to minimize a large number of patient condition codes in to a set of clusters. Existing methods, for example k-means clustering, cannot compress thousands of patient codes into a patient severity index. The prescription of antibiotic dataset has multiple prescriptions for a single patient ID; as a result more than one ICD-9 code is assigned to both patients and to antibiotics.

We will use text mining and clustering to group similar patients together so that a meaningful analysis can be performed. For example, we can study the difference in private insurance payments for antibiotic between the clusters, and a comparison can be made for analysis. Once again, we will use text mining to process and analyze the ICD-9 codes. For each patient for which antibiotic was prescribed, all the codes associated with multiple illnesses are combined into one text string.

For instance, when an antibiotic is prescribed for a patient, the physician who prescribes the antibiotic assigns the cause of the illness, which in turn defines a diagnosis code that is assigned to the patient. On another visit, the same patient might have a different prescription of antibiotic with the same ID and a different diagnosis code. What text mining does is to link all ICD-9 codes for the patient into one text string.

In this project, we will use text mining to reduce a large number of patient condition codes, which are denoted by ICD-9 codes. The ICD-9 codes are very large in number (in thousands). It is unrealistic to analyze each ICD-9 code, but by classifying them into reasonable clusters, we can study the probability distributions of each antibiotic on a variable of interest (for example, private insurance payments). For this project, we will classify the ICD-9 codes into six clusters to examine the relationship between different variables of interest; for example, total payment, private insurance payment, quantity of antibiotic, number of prescriptions, Medicare payment, and Medicaid payment.

Since there is more than one transaction per given day for a patient, many ICD-9 codes are assigned to the patient severity condition. It is impractical to analyze each ICD-9 code; as a result, classifying the ICD-9 codes into a reasonable cluster simplifies the analysis. We will use text mining to process and analyze the ICD-9 codes. For each prescription, all ICD-9 codes relating to multiple severity conditions are combined into one text string.

Discussion

For each prescription, there is a classification of disease, on which the ICD-9 code is used to denote the condition of the disease. As the number of prescriptions increases, the number of ICD-9 codes increases as well. Hence, classifying and studying each ICD-9 code for each transaction becomes cumbersome. We use clustering to define a small number of groups of documents so that documents within any one group are related and documents in different groups are not closely related.

The clustering algorithm uses a distance measure to classify observations in the same cluster; in most cases, the raw data are numerical. In contrast, text mining creates groups by looking at terms within each document. Documents or text within a group are represented by a list of terms, and those terms will appear in most of the documents within the group; a cluster is defined [27]. Once the observations or terms are classified into groups or clusters, a name will be assigned as a label that best describes the contents of the cluster or group. Domain knowledge is important when labeling a cluster, and as a result, we consult a pharmacist to get more information.

Chapter 7

ANTIBIOTICS RESULTS

We begin with exponential smoothing model building to forecast the prescription of antibiotics for dataset. As discussed in chapter 3, exponential smoothing models are characterized by giving heavier weights to recent events and lower weights to past events. Before we build a model for the prescription of antibiotics, we set the fit period and evaluation period. The fit period is the period on which the model fits the data; while the evaluation period is the period where we evaluate the model we built. We used a hold out sample of 20% of the total data set. By hold out sample, we mean we use 80% of the data to build a model and we use the remaining 20% to forecast for future values. For a better fit, it is better if a hold out sample is chosen.

As an example, we will build a model fit for the antibiotic, Amoxicillin. The private insurance payments made for the prescription of Amoxicillin will be forecast. Here, we are building a time series model to study the prescription practice of Amoxicillin. Plotting the forecasted private insurance payments vs. time gives the structure of the data, and gives a clue as to what kind of model might be appropriate for the dataset. The plot of private insurance payments vs. time

(Figure 7.4) shows how the series data grow or decay exponentially with a seasonal nature; as a result, we used exponential models. As discussed in chapter 3, there are several exponential smoothing models, so the question becomes which exponential model to fit. We used a simple exponential smoothing model, double (brown) exponential smoothing model, linear (holt) exponential smoothing model and damped trend linear exponential smoothing model.

We fitted several models and we chose the model with smallest root mean square error (RMSE) as a model of prediction. Statistically speaking, the model with the smallest error is considered the best. We begin the analysis of the private insurance payment with Amoxicillin. The prediction error plots for private insurance payments made for Amoxicillin are given in figure 7.1, while figure 7.2 describes the autocorrelation plots (autocorrelation, partial autocorrelation and inverse autocorrelation). Figure 7.3 describes a white noise plot and figure 7.4 describes the forecast plot.

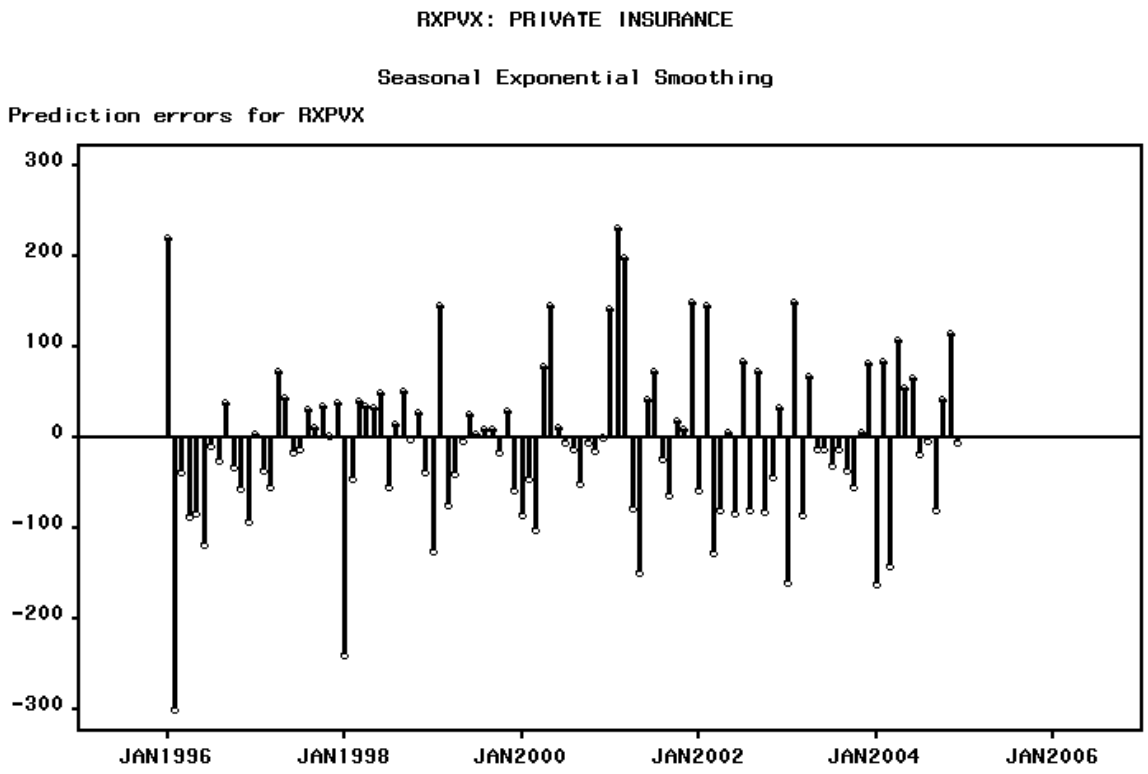

Figure 7.1 Prediction error Plots: Private Insurance Payment

The residuals don't appear to be white noise, with visual evidence of higher variability at the beginning of the time range. Figure 7.1 is the residual error plot for private insurance payments for the antibiotic, Amoxicillin.

The autocorrelation plot, figure 7.2, is within the bounds of 2 standard deviation errors, an indication that the residuals are white noise. The partial autocorrelation plots also reveal that the correlations are within 2 standard errors; an indication that the residuals are white noise.

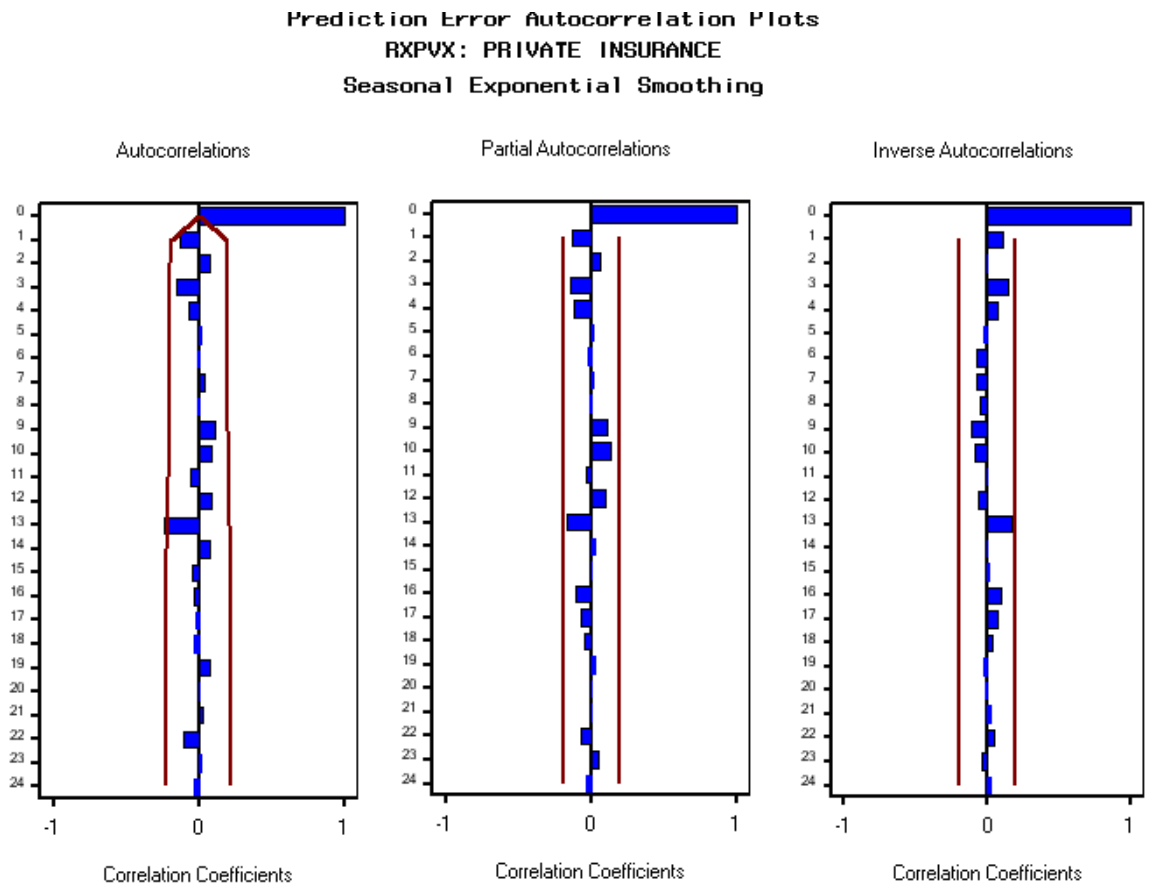

Figure 7.2 Prediction Error Autocorrelation Plots: Private Insurance Payments

We did also check white noise, unit root test and seasonal unit root tests that are given in figure 7.3. The white noise test indicates failure to reject a null hypothesis of white noise for alternative lags up to 24. The unit root tests indicate a rejection of a null hypothesis of a unit root for all polynomials up to lag 5, and the seasonal root tests indicate rejection of a null hypothesis of a seasonal unit root up to lag 5.

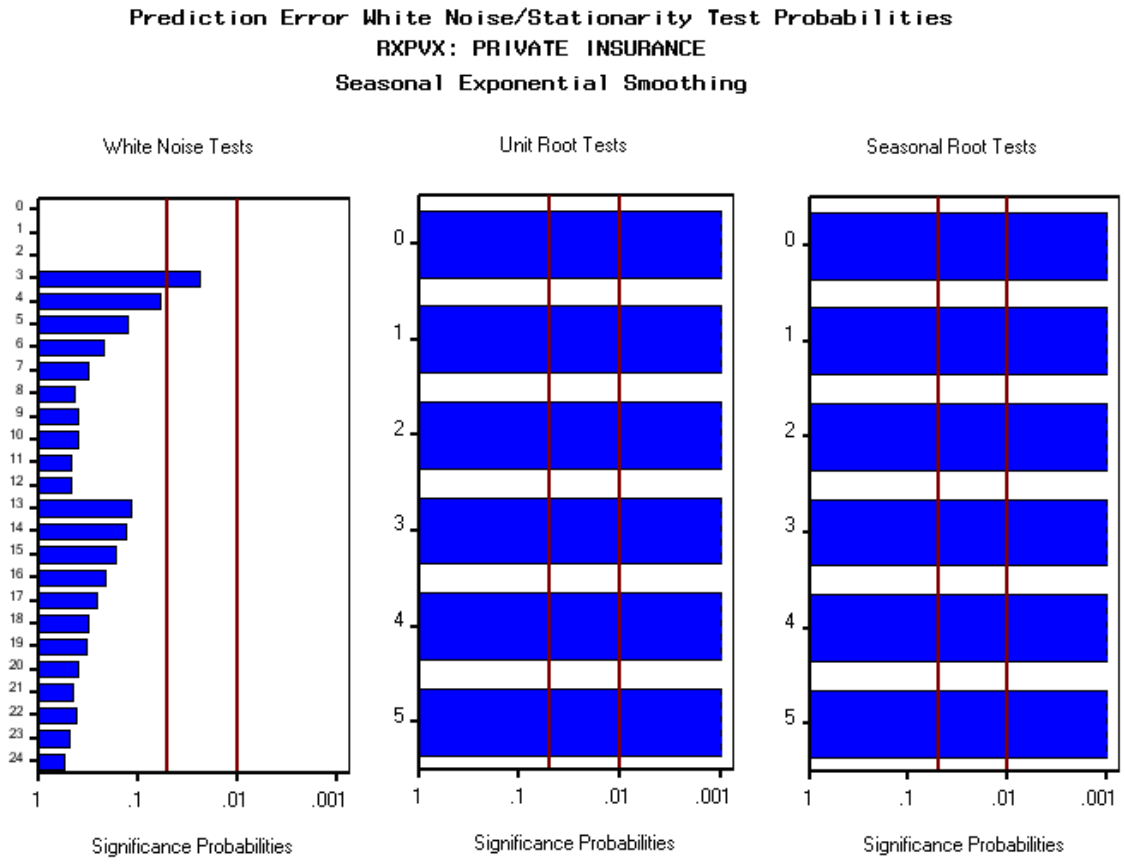

Figure 7.3 Prediction Error White Noise: Private Insurance Payments

We chose the best model for the private insurance payment made for the antibiotic, Amoxicillin, based on the smallest root mean square error (RMSE). We fitted an exponential smoothing model, a simple exponential smoothing model, double (brown) exponential smoothing model, linear (holt) exponential smoothing model and damped trend exponential smoothing model. From table 7.1, we selected the best model of fit for the private insurance payments made for the antibiotic, Amoxicillin.

We can see that the seasonal exponential smoothing model is selected as a model of forecast, with the smallest RMSE. The seasonal exponential smoothing model has also the smallest mean square error (MSE), mean absolute percent error (MAPE), mean absolute error (MAE) and largest Pearson correlation (R^2) time series.

Time series model	**MSE**	**RMSE**	**MAPE**	**MAE**	**R^2**
Seasonal exponential smoothing model	6831	82.649	32.19	59.679	0.508
Simple exponential smoothing model	8344.5	91.348	34.519	72.565	0.284
Double(brown) exponential smoothing model	10797.5	103.911	35.754	76.145	0.073
Linear Holt exponential smoothing model	8338.7	91.316	34.388	72.597	0.284
Damped trend exponential smoothing model	8599.2	92.732	34.664	73.141	0.262
Winters additive model	6898.6	83.058	32.513	68.925	0.408
Winters multiplicative model	12111.1	110.051	32.484	74.254	-0.040

Table 7.1 Exponential models and Statistics of Fit

Even though we used the RMSE for model selection in table 7.1, it is advisable to use MAPE for error interpretation since the RMSE value tends to be larger when compared to the MAPE. The MAPE in table 7.1 indicates that many cases have a percent error less than 32%.

The model estimate parameter of private insurance payment for the antibiotic, Amoxicillin is given in table 7.2.

Model Parameter	Estimates	Std. Error	T	Prob>|T|
LEVEL Smoothing weight	0.4230	0.0661	6.3971	<0.001
Seasonal Smoothing Weight	0.0010	0.1107	0.0090	0.9928
Residual Variance	7826			
Smoothed Level	339.3064			
Smoothed Seasonal Factor 1	205.1266			
Smoothed Seasonal Factor 2	99.9419			
Smoothed Seasonal Factor 3	106.0882			
Smoothed Seasonal Factor 4	-17.9435			
Smoothed Seasonal Factor 5	-67.4995			
Smoothed Seasonal Factor 6	-86.7825			
Smoothed Seasonal Factor 7	-77.4002			
Smoothed Seasonal Factor 8	-69.2203			
Smoothed Seasonal Factor 9	-33.8578			
Smoothed Seasonal Factor 10	-35.2863			
Smoothed Seasonal Factor 11	-49.4234			
Smoothed Seasonal Factor 12	26.1430			

Table 7.2 Parameter Estimates of Private Insurance Payment: Amoxicillin

The smoothed seasonal factor one is for January. The exponential smoothing model detects a seasonal difference decrease of $105.18 (99.94191-205.12659=105.18468) for February compared to January for private insurance payments made. On the other hand, the smoothed seasonal factor eleven indicates November; the exponential smoothing model detects a seasonal difference increase of $75.56 (26.14302-(-49.42344)=75.566) for December compared to November private insurance payments made.

We have shown so far that forecast values for private insurance payments of Amoxicillin were fit by the exponential smoothing model. The seasonal exponential model has multiple steps ahead prediction, and table 7.2 gives up to a smoothed seasonal factor level 12; this is an indication that seasonal data forecast over a longer time period will be more accurate than forecasts over a short period of time. Here, we are analyzing the private insurance payments made that were obtained by summing daily expenses paid for a period of every month.

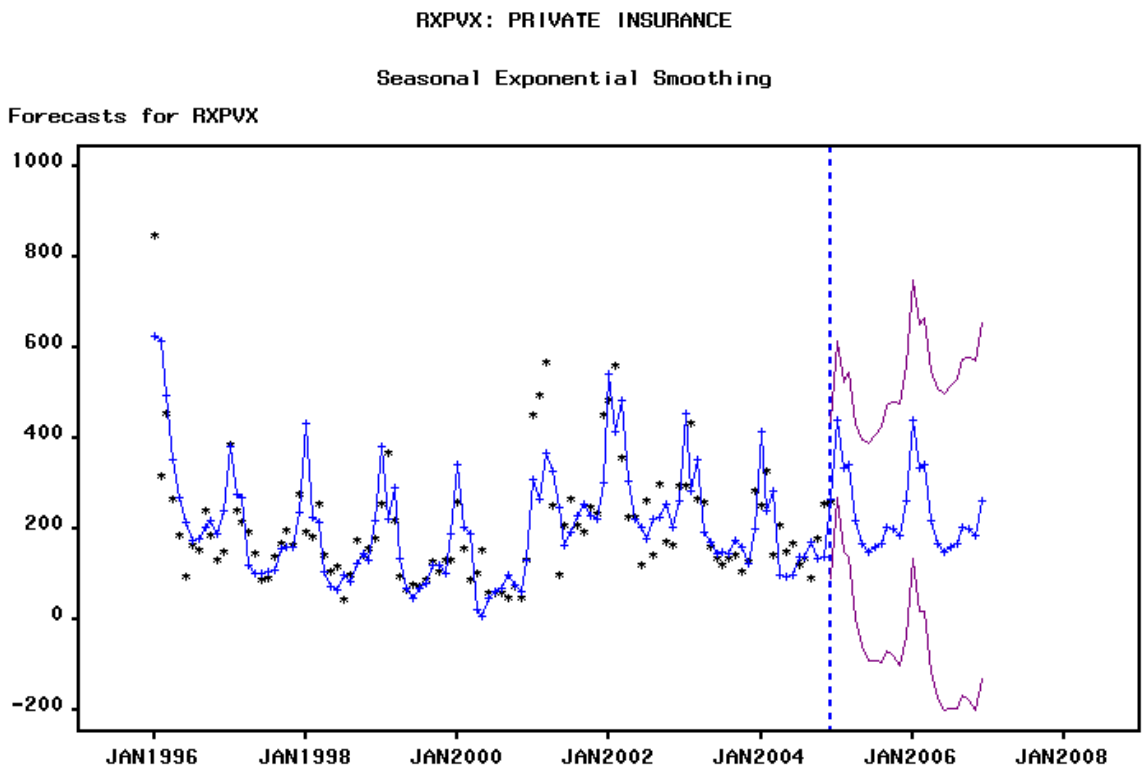

Figure 7.4 Forecast for Private Insurance Payments: Amoxicillin

The forecast plot for private insurance payments made for Amoxicillin shows seasonality with a period of approximately 12 months, which supports the model chosen. Figure 7.4 reveals that predicted and actual private insurance payments are very close to each other, making the residual term very small.

The predicted plot for private insurance payment for the antibiotic, Amoxicillin, also indicates that the values are higher around January and lower around June; this event repeats at approximately 12 months. We observe that the predicted series is very close to the actual series, which is an indication that the model that we built fits the data well. We have used the historical data to build what the

private insurance payment will be in the future. One has to be cautious in forecasting for a longer period; in the future, the forecast might not be as accurate as we expect. For this reason, we forecast what the private insurance payments might be for the next two years, i.e. for the years 2005-2006.

We have also investigated on average how much patients are spending on prescriptions. At this time, we predict the total payments made on antibiotics instead of private insurance payments made.

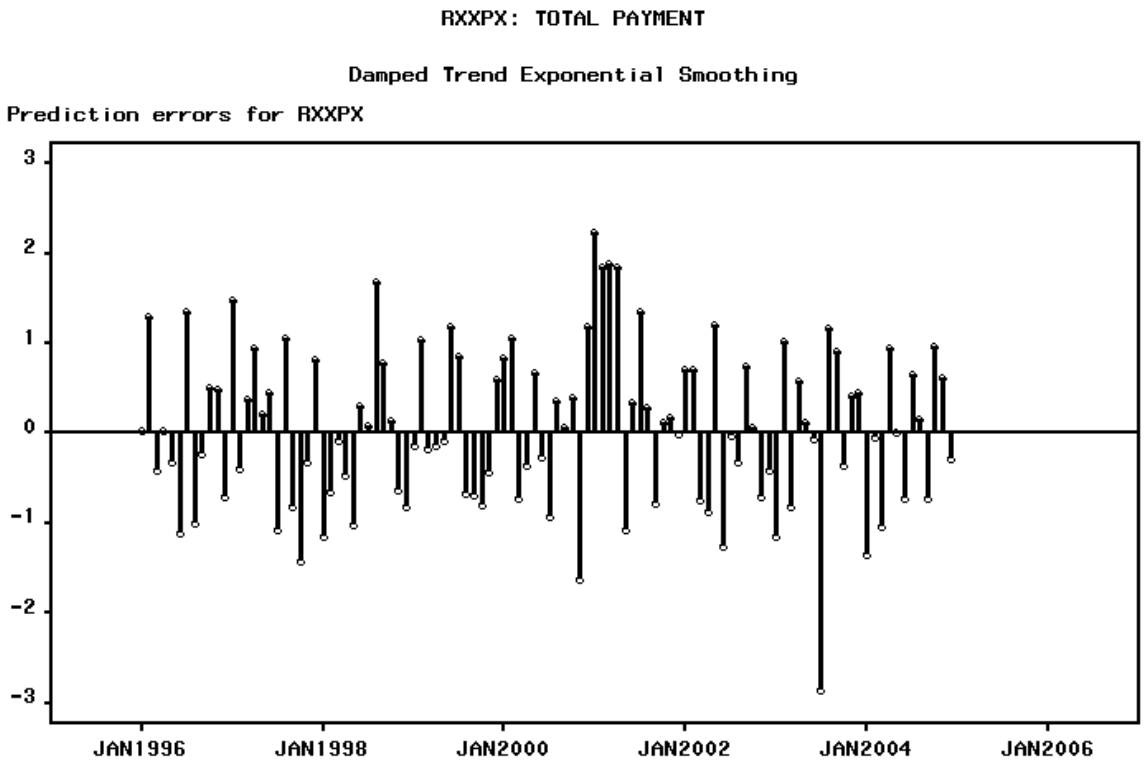

Figure 7.5 Prediction Error Plots: Total Payments: Amoxicillin

The residuals do not appear to be white noise, with visual evidence of higher variability at the middle and at the end of the time range. Figure 7.5 is the residual error for total insurance payment for the antibiotic, Amoxicillin.

The autocorrelation plot, figure 7.6, is within the bounds of 2 standard deviation errors, an indication that the residuals are white noise. The partial autocorrelation plots also reveal that the correlations are within 2 standard errors; an indication that the residuals are white noise.

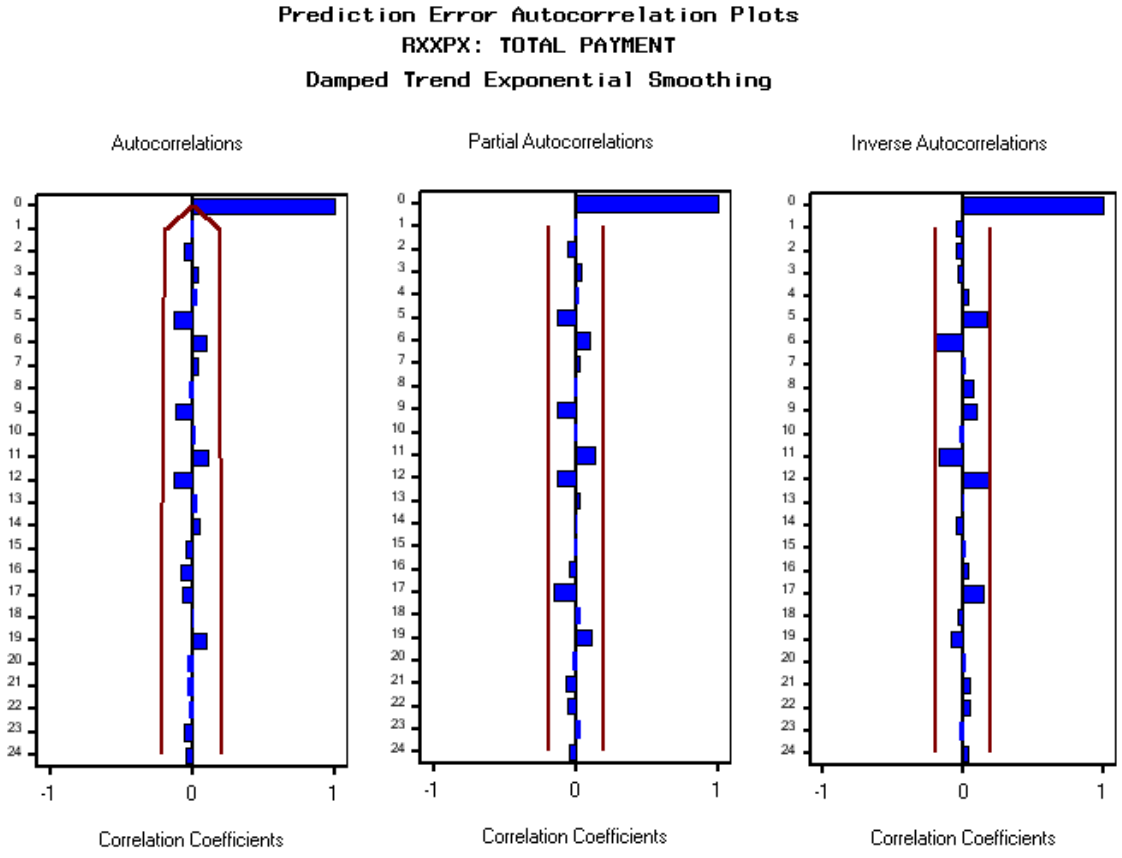

Figure 7.6 Prediction Error Autocorrelation Plots: Total Payments

We also checked white noise, unit root test and seasonal unit root tests, which are given in figure 7.7. The white noise test indicates failure to reject a null hypothesis of white noise for alternative lags up to 24. The unit root tests indicate a rejection of a null hypothesis of a unit root for all polynomials up to lag 4, and the seasonal root tests indicate rejection of a null hypothesis of a seasonal unit root up to lag 5.

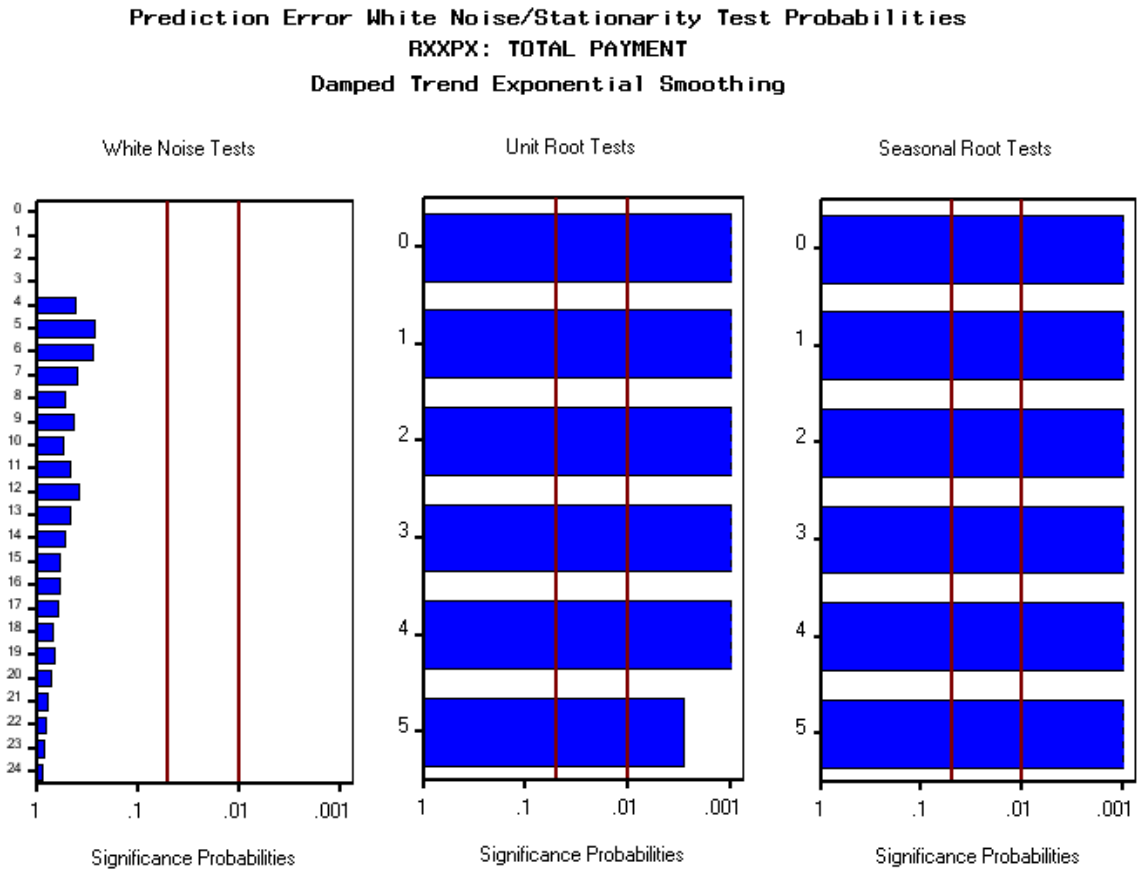

Figure 7.7 Prediction Error White Noise: Total Payments: Amoxicillin

We chose the best model for the total payment made for the antibiotic, Amoxicillin, based on the smallest root mean square error (RMSE). It should be

noted that here we are investigating how much patients on average are paying for the antibiotic, Amoxicillin. We fitted a seasonal exponential smoothing model, simple exponential smoothing model, double (brown) exponential smoothing model, linear (holt) exponential smoothing model and damped trend exponential smoothing model. From table 7.3, we selected the best model of fit for the total payments made for the antibiotic, Amoxicillin. We can see that the damped trend exponential smoothing model is selected as a model forecast, with the smallest RMSE. The damped trend exponential smoothing model has also the smallest mean square error (MSE), mean absolute percent error (MAPE), mean absolute error (MAE) and largest Pearson correlation (R^2) time series.

Time series model	MSE	RMSE	MAPE	MAE	R^2
simple exponential smoothing model	0.4943	0.7031	27.4015	0.5496	-0.094
Double(brown) exponential smoothing model	0.5244	0.7242	28.2752	0.5817	-0.160
Seasonal exponential model	0.5145	0.7173	28.4545	0.5672	-0.138
Linear Holt exponential smoothing model	0.4968	0.7049	27.8328	0.5549	-0.099
Damped trend exponential smoothing model	0.4943	0.7030	27.4014	0.5495	-0.094
Winters additive model	0.5207	0.72163	28.84722	0.5726	-0.152
Winters multiplicative model	0.5257	0.72506	29.10547	0.5694	-0.163

Table 7.3 Exponential models and Statistics of fit

Even though we used RMSE for model selection in table 7.3 as mentioned in the previous paragraph, it is advisable to use MAPE for error interpretation because the RMSE value tends to be larger when compared to the MAPE. The MAPE in table 7.3 indicates that many cases have a percent error less than 27%. The model estimate parameter of total payment for the antibiotic Amoxicillin is given in table 7.4.

Model Parameter	Estimate	Std. Error	T	Prob >|T|
Level smoothing weight	0.24354	0.0722	3.3717	0.0019
Trend smoothing weight	0.07557	0.1038	0.7280	0.4717
Damping smoothing weight	0.99106	0.1746	5.56775	<0.001
Residual variance	0.81101			
Smoothed level	12.12410			
Smoothed trend	0.12634			

Table 7.4 Parameter Estimates of Total Payment on Average: Amoxicillin

The t-test p-value in table 7.4 may be questionable due to the fact that the trend smoothing weight falls on the boundary of zero estimation bounds.

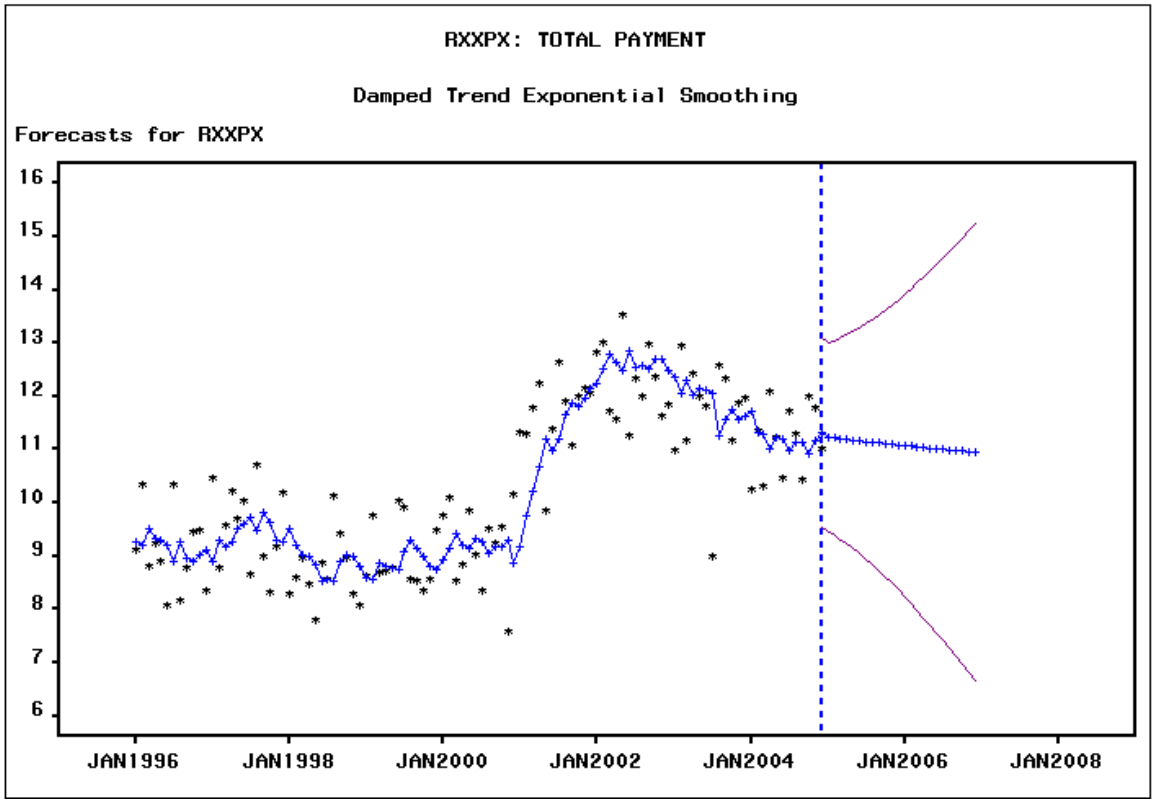

Figure 7.8 Forecast for Total Payments: Amoxicillin

The forecast plot for total payments made for Amoxicillin shows a slightly increasing trend from January, 2001 up to June, 2002 with damping seasonality. Figure 7.8 reveals that predicted and actual total payments made are very close to each other, making the residual term very small.

The predicted plot for total payment for the antibiotic, Amoxicillin, also indicates that the values are higher around January and lower around June; this event repeats at approximately 6 months. We also observe that the predicted series is very close to the actual series, which is an indication that the model is a good fit. We used the historical data to build what the total payment will be in the future.

Erythromycin Results

In this section of the analysis, we will talk about autoregressive models (AR), moving average models (MA), autoregressive moving average models (ARMA) and autoregressive integrated moving average models (ARIMA). We will build models for the antibiotic, Erythromycin, prescription. We use this example in contrast to the use of Amoxicillin in the previous section because the optimal model is ARIMA in contrast to exponential smoothing.

In chapter four, we have seen that an autoregressive process of order p is a linear function of p past values plus an error term. We have shown in the previous section that if an autoregressive model doesn't have a constant mean and variance through the sequence of time, there is a need for differencing.

As a result, an autoregressive integrated moving average (ARIMA) model is introduced to solve the problem of non stationarity. What an ARIMA model does is to take the difference of the series of data to make the series stationary.

As a start for the analysis of the antibiotic, Erythromycin, we plotted the total payment against the start date of antibiotics; this plot is given in figure 7.9. We will fit an autoregressive model of order p, i.e., AR(p). Based on the smallest MAPE, we will select p. As a result, the AR(p) model that fits the data well will be selected. We will analyze the fit of the data by checking autocorrelation plots, white noise and stationarity test, and the forecasted plot.

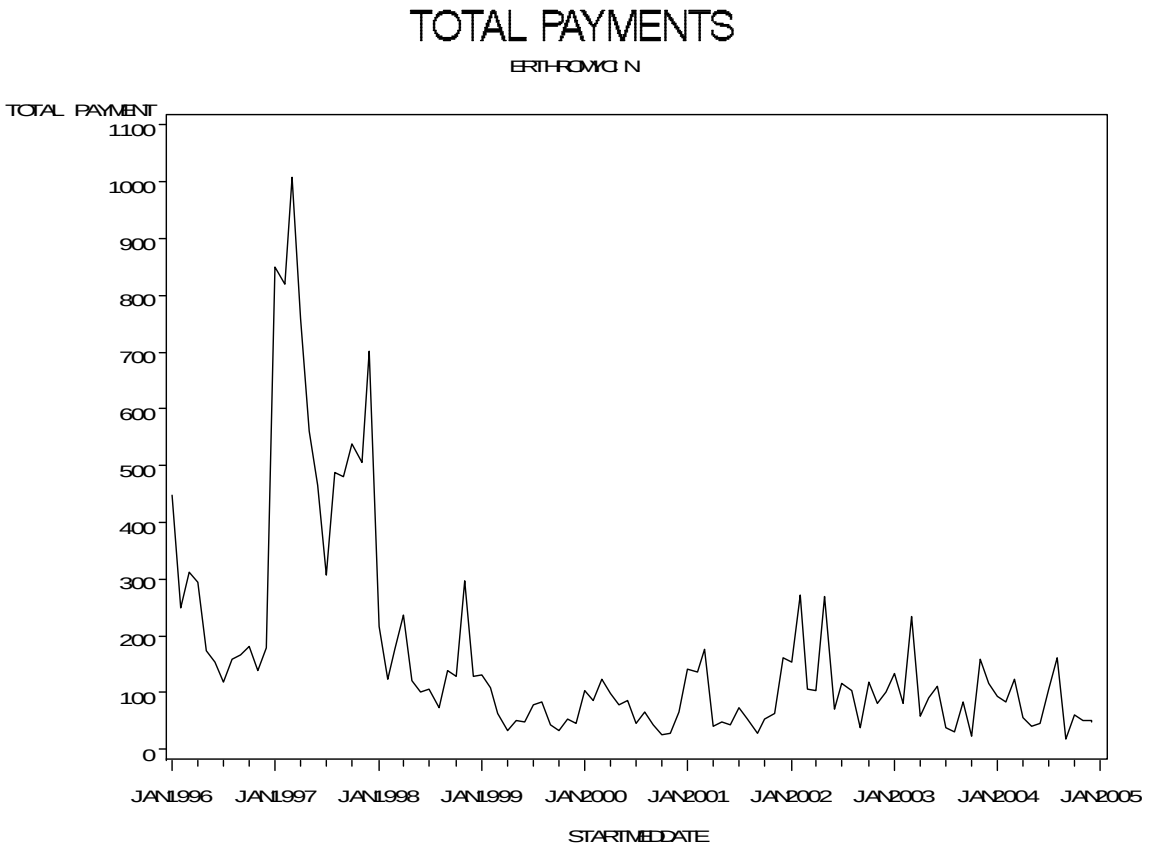

Figure 7.9 Plot of Total Payments vs. start date: Erythromycin

Figure 7.9 differs from that of figure 7.4 for Amoxicillin in the previous section in that there is variability in the payments made from January, 1997 through January, 1998, but for the case of Amoxicillin there is seasonal variability with exponential increasing in the payments made.

We first investigated whether the series of data is generated from a white noise; to do this, we plotted the total payments made versus the start date of antibiotic, given in figure 7.9. Visual inspection shows the series is not generated from white noise, but visual inspection is not enough to conclude whether the series is generated from a white noise or not. As a result, we will check the residual plot, autocorrelation plot, white noise stationarity test and see how the model built fits the data.

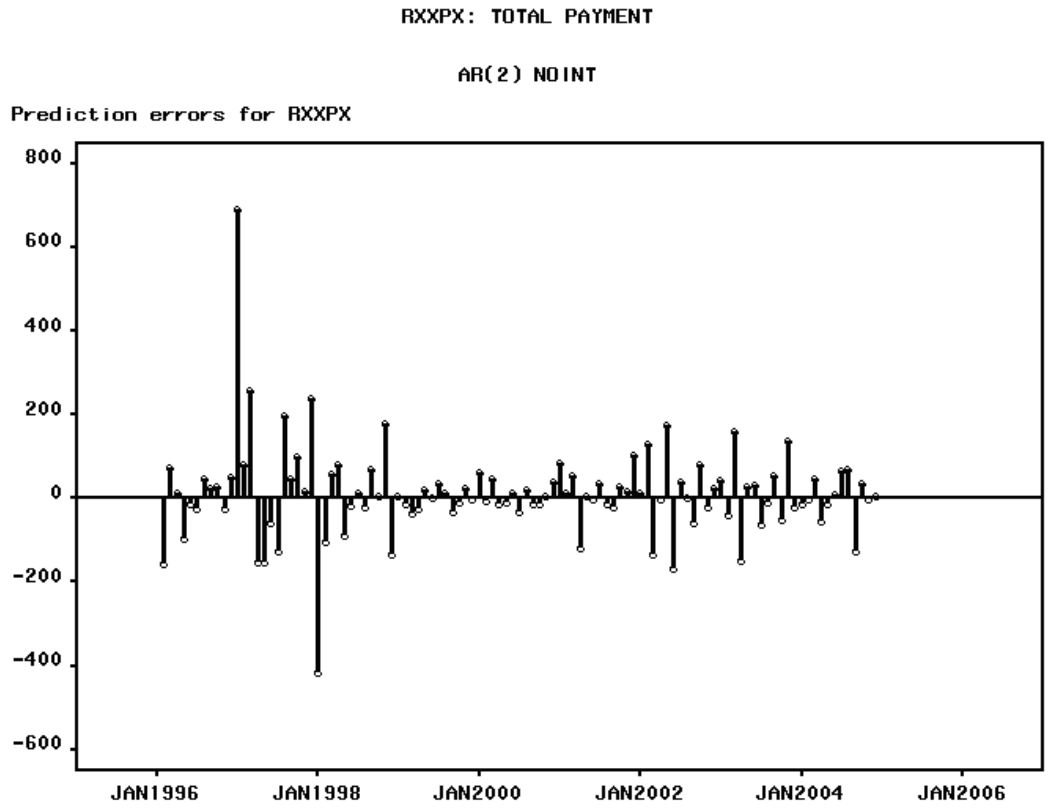

Figure 7.10 Prediction Error Plots: Total Payments: Erythromycin

The residual plot given in figure 7.10 does not appear to be white noise, even though there is some variability at the beginning of the series. Figure 7.10 is the residual error for total payments made for the antibiotic, Erythromycin. Figure 7.10 differs from that of figure 7.5 in the previous section in that the residual is smaller and less variable in the distribution of the error terms.

The autocorrelation plot, figure 7.11, is within the bounds of 2 standard deviation errors, an indication that the residuals are white noise, even though at lag twelve, the inverse autocorrelation is non-significant; that is, outside the bound of 2

standard deviation errors. The partial autocorrelation plots also reveal that the correlations are within 2 standard errors; an indication that the residuals are white noise.

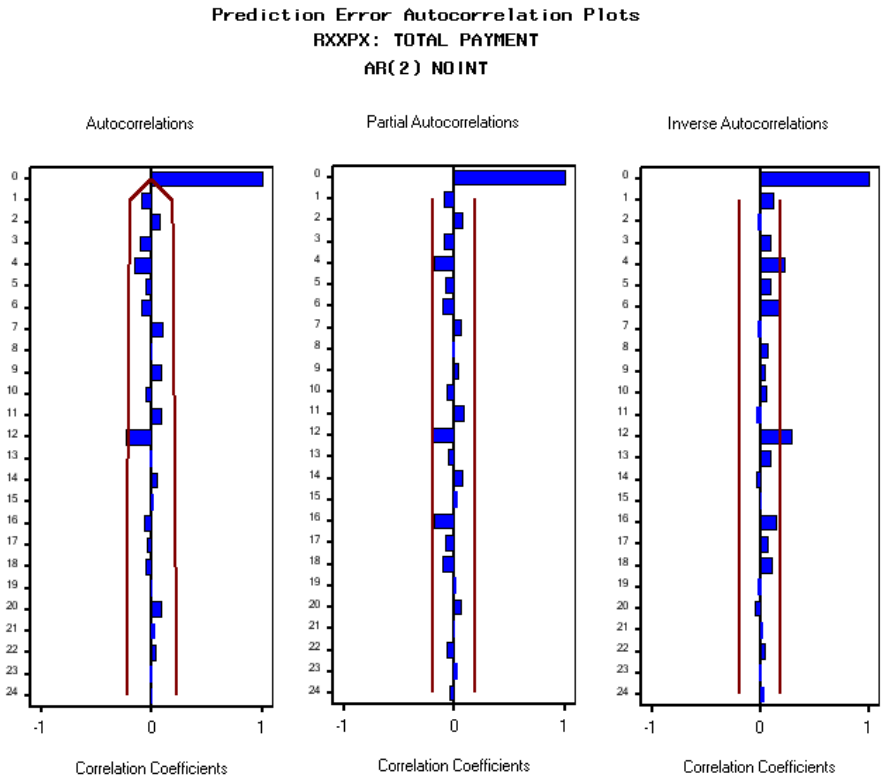

Figure 7.11 Prediction error Autocorrelation Plots: Total Payments

Figure 7.11 differs from that of figure 7.6 for Amoxicillin in the previous section since all lags are not significant while the inverse autocorrelation function value at lag 12 is marginally significant.

We also checked white noise, unit root test and seasonal unit root tests, which are given in figure 7.12. The white noise test indicates failure to reject a null

hypothesis of white noise for alternative lags up to 24. The unit root tests indicate a rejection of a null hypothesis of a unit root for all polynomials up to lag 5, and the seasonal root tests indicate rejection of a null hypothesis of a seasonal unit root up to lag 5.

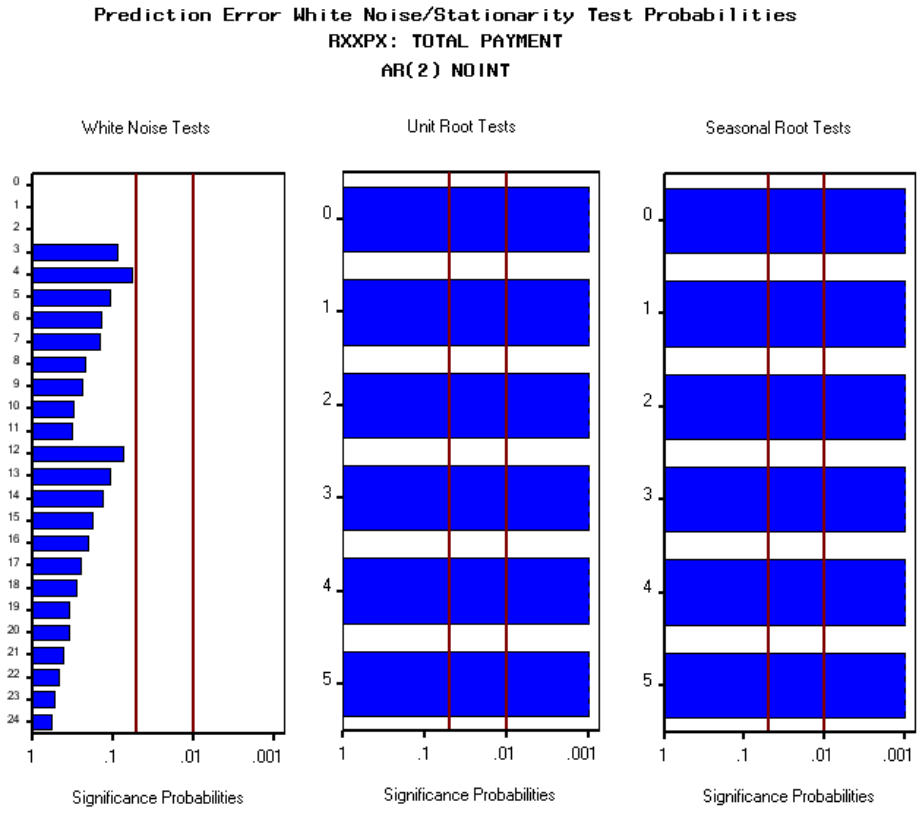

Figure 7.12 Prediction error White Noise: Total Payments: Erythromycin

Figure 7.12 differs from that of figure 7.7 for Amoxicillin in that the unit root tests indicate a rejection of a null hypothesis of a unit root for all polynomials up to lag 5 while in figure 7.7, the unit root tests indicates a rejection of a null hypothesis of a unit root for all polynomials up to lag 4.

We chose the best model for the total payment made for the antibiotic, Erythromycin, based on the smallest root mean square error (RMSE). It should be noted that here we are investigating how much patients are paying in total for the antibiotic, Erythromycin. The autoregressive order two, AR(2) was a perfect fit for the data and the parameter estimates of the model are given in table 7.5. We can see that the AR(2) model is selected as a model of forecast, with the smallest RMSE. The MAPE in table 7.5 indicates that many cases have a percent error less than 27%.

Time series model	**MSE**	**RMSE**	**MAPE**	**MAE**	**R^2**	**AIC**	**SBIC**
AR(1)	6487.7	80.546	28.985	59.98	-0.738	317.996	319.579
AR(2)	6075	77.942	27.45	57.758	-0.627	317.629	320.797
AR(3)	6694.1	81.817	32.856	63.131	-0.793	325.123	331.457
AR(4)	7152.7	84.574	33.784	65.089	-0.916	327.509	333.843
AR(5)	6505.7	80.658	34.765	61.600	-0.743	326.096	334.013

Table 7.5 Autoregressive models and Statistics of fit

We have mentioned in the analysis of the antibiotic, Amoxicillin, that the RMSE was very large when compared to the MAPE; but for the antibiotic, Erythromycin, the RMSE value is not that big as we can see from table 7.5. For error

interpretation, the MAPE is better, and from table 7.5, the MAPE indicates that many cases have a percent error less than 27%. The model estimate parameter of total payment for the antibiotic, Erythromycin is given in table 7.6.

Model Parameter	Estimate	Std. Error	T	Prob >|T|
Autoregressive, Lag 1	0.8536	0.118	7.213	<.0001
Autoregressive, Lag 2	0.0629	0.118	0.531	0.598
Model Variance	15839			

Table 7.6 Parameter Estimates of Total Payment: Erythromycin

Even though the parameter estimate for autoregressive lag 2 in table 7.6 is non-significant, an autoregressive model of order 2, AR(2) was perfectly fit to the data.

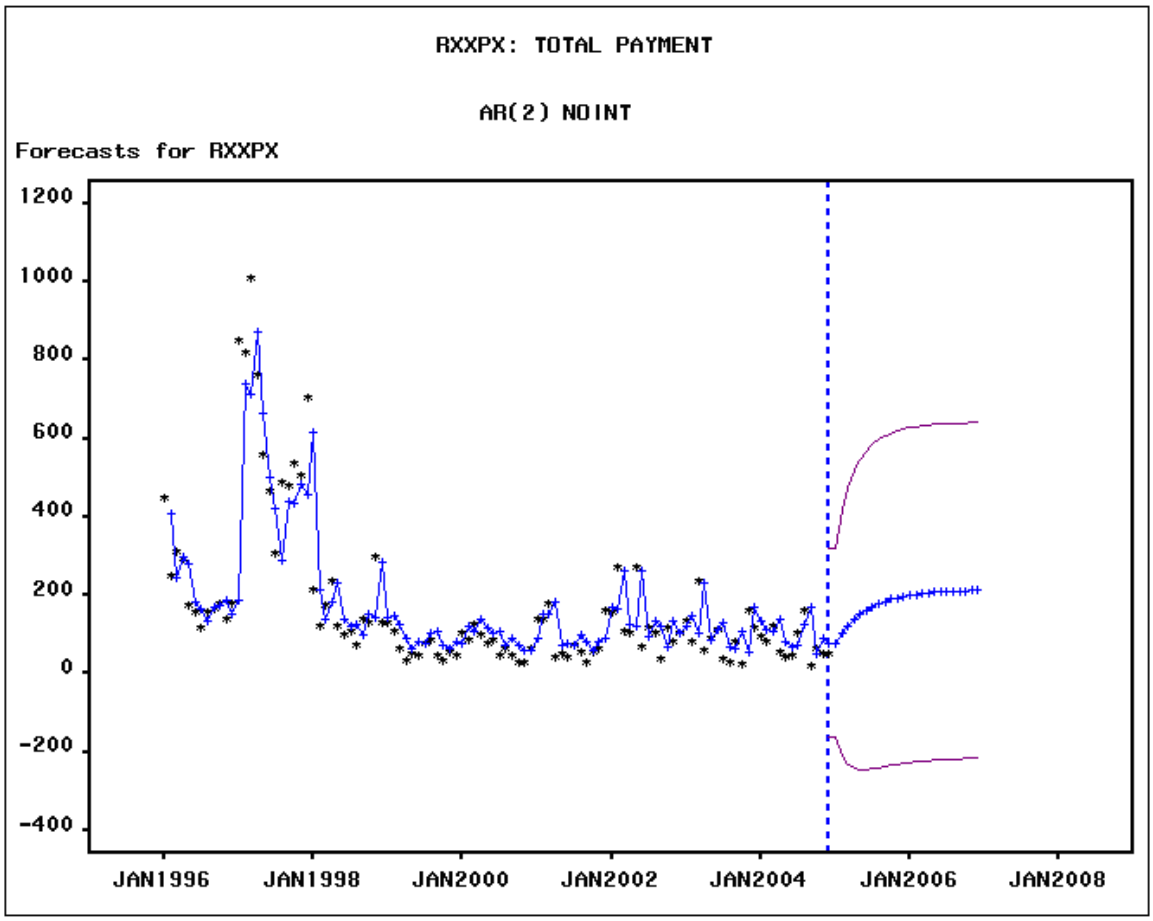

Figure 7.13 Forecast for Total Payments: Erythromycin

The forecast plot for total payments made for Erythromycin shows a slightly decreasing trend starting January, 1998 and peaks starting January, 2005. In contrast, the forecast for Amoxicillin shown in figure 7.8 is exponentially increasing through the range of time. Figure 7.13 reveals that predicted and actual total payments made are very close to each other; except for values between June, 1997 and January, 1998; there is a small difference between the actual and forecasted values.

The predicted plot for total payment for the antibiotic, Erythromycin, also indicates that the values are higher around January and lower around June; this event repeats at approximately 6 months, and is similar to that for Amoxicillin. We also observe that the predicted series is very close to the actual series, which is an indication that the model is a good fit. So far from our analysis examples, the cost analysis of antibiotics has a seasonal nature. By seasonal nature, we mean that an event happens repeatedly; for example, if the event happens every six months, we say it has six-month seasonality.

High Performance Forecasting

In this section of the chapter, we will use the High Performance Forecasting procedure to forecast prescriptions of antibiotics. The High Performance Forecasting analyzes time series (observations that are equally spaced at a specific time interval, for our case, month), or transactional data (observations that are not equally spaced with respect to a particular time interval as in our original antibiotic dataset). We will also plot the forecast series for several antibiotics in the same plot; in that case, comparisons can be made very easily. The historical series and forecasted series for several antibiotics can be drawn in the same plot, so that comparisons can be made and further studies can be performed on the nature of the trend of the forecast. In this section, we will identify and forecast series components such as the actual, predicted, lower confidence interval, upper confidence limit, prediction error, trend, seasonality

and error (irregular). Trend usually refers to a deterministic function of time, which means the deterministic component exhibits no random variation and can be forecast perfectly, while a stochastic component is subject to random variation and can never be predicted perfectly except for chance occurrence.

Seasonality refers to the repetitive behavior at known seasonal periods, for instance, six months for antibiotics prescription. The High Performance Forecasting splits the current value of observation into trend, seasonal and error Components. In contrast to the previous section on time series forecasting, high performance forecasting can be used to forecast several variables by group, determines the best model from a list of models using the technique HPF DIAGNOSE and HPF ENGINE and can be used to forecast for a database with a large number of observations.

In this analysis section of the dissertation, we will use the antibiotic, Cipro, as an example to show and forecast the variables of interest (such as total payment, private insurance payment, quantity, amount of prescription, Medicare and Medicaid). As mentioned in chapter two, the data analysis section, we have created an antibiotics dataset by concatenating each antibiotic. Cipro is a commonly prescribed antibiotic, and the dataset we have shows a tremendous number of observations. Figure 7.14 gives the plot of total payment, private insurance payment, Medicare payment and Medicare against the start date of antibiotic. From this plot, we can visually inspect how much difference really

occurs between total payment, private insurance payment, Medicare payment and Medicare payments and give ideas as to what to expect when we make the forecast for these variables. We have used SAS CODE 7 to create the Cipro dataset and plot the figure 7.14.

```
Data cipro; set diser.disertation;
where RXNAME IN ('CIPRO');
RUN;

PROC SORT DATA=cipro;
by RXNAME STARTMEDDATE;
run;

Proc hpf data=cipro out=cipro lead=0;
id startmeddate interval=month accumulate=total;
forecast RXMDX RXMRX RXPVX RXXPX /model=none;
run;

title1 'Cipro Variables';
axis2 label=(a=-90 r=90 "PAYMENT");
SYMBOL1 INTERPOL=JOIN HEIGHT=10pt VALUE=NONE CV=BLUE
LINE=1 WIDTH=2;
SYMBOL2 INTERPOL=JOIN HEIGHT=10pt VALUE=NONE CV=GREEN
LINE=1 WIDTH=2;
SYMBOL3 INTERPOL=JOIN HEIGHT=10pt VALUE=NONE CV=RED
LINE=1 WIDTH=2;
SYMBOL4 INTERPOL=JOIN HEIGHT=10pt VALUE=NONE CV=CYAN
LINE=1 WIDTH=2;
Legend1 FRAME;
Axis1 STYLE=1 WIDTH=1 MINOR=NONE;

PROC GPLOT DATA=CIPRO;
PLOT RXMDX*STARTMEDDATE RXMRX*STARTMEDDATE
RXPVX*STARTMEDDATE
         RXXPX*STARTMEDDATE /OVERLAY HAXIS=AXIS1
VAXIS=AXIS2
         HAXIS='01JAN1996'D TO '01JAN2005'D BY YEAR
       FRAME LEGEND=LEGEND1;
RUN;
```

SAS CODE 7

Table 7.7 SAS codes for model forecasting and plotting: Cipro

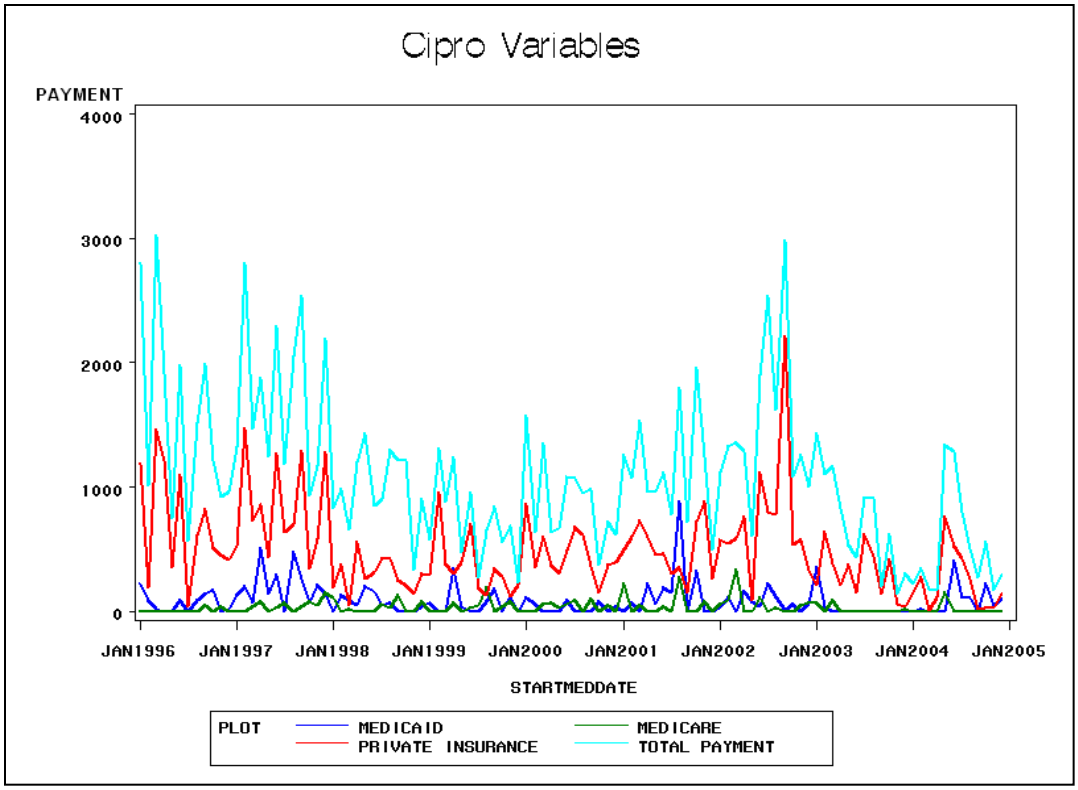

Figure 7.14 Cipro variables plot vs. Start Date of Antibiotics

We have used SAS CODE 7 to accumulate the daily observations into an accumulated sum of monthly values, and then plotted each variable of interest (Medicaid, Medicare, Private Insurance and Total Payment) against the start antibiotic date. Figure 7.14 gives a starting point to see the relationships between these variables, and we can easily see that the Total Payment made surpasses Medicaid, Medicare and Private Insurance payments. On the other hand, the Medicare payment made for Cipro is smaller when compared to the Medicaid for almost every time point. As we can see from figure 7.14, the plot for insurance payment fluctuates randomly; this might indicate that the data for Cipro

is non stationary. As a result, an ARIMA model is one of the choices as a model prediction.

Also from figure 7.14, we can see that between April, 2002 and July, 2002, there is a sudden increase in the insurance payment made for Cipro. We can think of this as level shift or an event that happened at that moment of time that makes the increase, or that particular observation could just be an outlier. The model that we will build automatically detects whether a particular observation is an outlier or not. The dataset, Cipro, has 1674 daily observations obtained from the means procedure given in table 7.8. Cipro is a commonly prescribed antibiotic, and we have a reasonable number of observations for model building and forecasting; as a result, we will use it as an example to build a model for prediction. The Proc Means SAS procedure given in SAS CODE 8 was used to analyze the distribution of the observations. Once we have an idea how the observations are linked, we can select a model based on the number of observations, maximum, minimum and standard deviation of the antibiotics dataset.

The MEANS Procedure

Analysis Variable: RXPVX = PRIVATE INSURANCE

ANTIBIOTICS NAME	N Obs	Sum	Minimum	Maximum	STDV
AMOXICILLIN	20457	44292.52	0.00	111.88	4.61
AMPICILLIN	1038	1693.86	0.00	54.60	3.52
AZITHROMYCIN	377	1985.20	0.00	262.27	18.45
CEFACLOR	597	14392.84	0.00	187.05	30.07
CEFADROXIL	239	5741.64	0.00	181.00	28.61
CEFUROXIME	28	2039.83	0.00	156.80	41.52
CEPHALEXIN	4788	41841.26	0.00	124.10	13.69
CIPRO	1674	53009.37	0.00	323.45	44.45
CLARITHROMYCIN	156	4785.03	0.00	87.38	35.39
CLINDAMYCIN	403	4597.68	0.00	123.40	16.53
CLOTRIMAZOLE	647	2210.43	0.00	142.11	14.63
DICLOXACILLIN	487	1277.08	0.00	49.24	8.06
DOXYCYCLINE	1054	8482.74	0.00	121.12	14.96
ERYTHROMYCIN	2226	5077.73	0.00	62.16	5.60
KEFLEX	1490	27802.95	0.00	115.78	29.37
SULFAMETHOXAZOLE	113	561.46	0.00	19.00	8.10
TEQUIN	408	11831.47	0.00	174.75	33.81
TETRACYCLINE	477	1395.82	0.00	97.48	10.21
TOBRAMYCIN	839	4003.18	0.00	44.69	5.16
VANCOMYCIN	73	703.34	0.00	262.60	34.17

Table 7.8. The means procedure for Private Insurance

```
Proc means data=Diser.Disertation sum nway min
max std maxdec=2;
        class RXNAME;
        VAR RXPVX;
RUN;
```

SAS CODE 8

Table 7.9 SAS code for the Means procedure

From the means procedure given in table 7.8, we selected the antibiotic, Cipro, to investigate the private insurance payments made. As the main purpose of this dissertation is to build model forecasts for antibiotics, building model forecasts for Cipro's private insurance payment would be an ideal situation.

We will first build several models using the Proc hpfdiagnose statement shown in SAS CODE 9, and put the models into the model repository (warehouse). We then utilize the models built in hpfdiagnose to the dataset using hpfengine. These methods select the best model using the criterion specified; for our case, we selected RMSE as an error of model selection. The model repository is a collection of a set of models to use to forecast the antibiotic, Cipro. The HPFDIAGNOSE procedure builds a model based on MODELREPOSITORY, and models specified such as ARIMAX (autoregressive moving integrated moving average model), ESM (exponential smoothing model) and UCM (unobserved components model). The HPFENGINE procedure selects the models based on the smallest RMSE, and plots several graphs such as forecast, residual, autocorrelation plot etc.

We have built a model to forecast the private insurance payment made for the antibiotic, CIPRO. The parameter estimates given in table 7.9 indicate that there is an outlier on September, 2002; we have visually estimated the range of points where an outlier might happen as discussed previously. The plot of private insurance payments vs. start antibiotic date given in figure 7 14 gives a starting point to show how the data behave through time. We can see that there is fluctuation of the data points across different time points, which we suspect indicates non-stationarity. For this reason, an ARIMA model would be a typical model of choice. We built a predictive model using the SAS CODE 9 for private insurance payments made for the antibiotic, Cipro.

Parameter Estimates					
Component	**Parameter**	**Estimate**	**Standard Error**	**t Value**	**Approx Pr > |t|**
RXPVX	**AR1_1**	-0.52158	0.09584	-5.44	<.0001
RXPVX	**AR1_2**	-0.26791	0.09718	-2.76	0.0069
AO01SEP2002D	**SCALE**	1700.4	351.03913	4.84	<.0001

Table 7.10 Parameter Estimates of Private Insurance Payment: Cipro

The component in the parameter estimate given in table 7.10 suggests that the event in September, 2002 resulted in an increase of $1700.40 in the payment of private insurance for the antibiotic, Cipro. We also have significant AR(1) and AR(2) parameters that indicate the model is well built. The parameter, AR1_1, of the component, RXPVX (Private Insurance payment), represents an autoregressive process of order 1 with a difference of one, while AR1_2 is an

autoregressive process of order 2 of difference one. The parameter, AO01SEP2002D, represents an additive outlier at September, 2002 and the scale indicates that the increase occurred due to the additive outlier.

```
PROC HPFDIAGNOSE DATA=CIPRO

        OUTEST=CIPROSTATE CRITERION=RMSE
        BASENAME= AMXESM PRINT=SHORT
        MODELREPOSITORY=SASUSER.ANTIBIOTICSMODELS;
ID STARTMEDDATE INTERVAL=MONTH;
FORECAST RXPVX ;
ARIMAX PERROR=(12:24) P=(0:12) Q=(0:12) CRITERION=SBC
METHOD=MINIC;
ESM; UCM;
RUN ;ODS RTF;
ODS GRAPHICS ON;
PROC HPFENGINE DATA=CIPRO

MODELREPOSITORY=SASUSER.ANTIBIOTICSMODELS

INEST=CIPROSTATE

                        GLOBALSELECTION=TSSELECT
                        PRINT=(SELECT ESTIMATES) LEAD=24
                        OUTFOR=CIPROTOTAL
                        OUTEST=CIPROTEST
                        OUTSTAT=CIPROSTAT PLOT=ALL;
                FORECAST RXPVX;
        ID STARTMEDDATE INTERVAL=MONTH;
RUN;
ODS RTF CLOSE;
ODS GRAPHICS OFF;                SAS CODE 9
```

Table 7.11 SAS code for building Private Insurance model for Cipro

The HPFDIAGNOSE procedure builds models from the selected list (ARIMAX, ESM and UCM); the MODELREPOSITORY is a warehouse for the model built and the model estimates are put in the OUTEST=dataset. The HPFENGINE procedure builds the best model from the MODELREPOSITORY, which also uses some models from GLOBALSELECTION= where the new models built are stored in MODELREPOSITORY=SASUSER.ANTIBIOTICS. The statements, OUTFOR= OUTEST= and OUTSTAT=, put the values of the forecast, estimates and statistics respectively in a dataset.

The PERROR= specifies the range of the AR order for obtaining the series, P=specifies the AR order, Q= specifies the range of the MA order, CRITERION=SBC specifies that Swartz Bayesian Criterion is selected and the METHOD=MINIC specifies Minimum Information Criterion is selected. We have selected the best model for Cipro private insurance payment from the listed models in SAS CODE 9. The SAS CODE 9 selects the best model of forecast based on the smallest RMSE, and from table 7.12, we see that an ARIMA model with autoregressive order two and difference of one is chosen as the predictive model. The model AMXESM50, AMXESM 51 and AMXESM 52 are the models that were selected from the list of models in the model repository.

Model Selection Criterion = RMSE			
Model	**Statistic**	**Selected**	**Label**
AMXESM50	398.24401	Yes	ARIMA: RXPVX ~ P = (1,2) D = (1) NOINT
AMXESM51	428.92004	No	Simple Exponential Smoothing
AMXESM52	511.79995	No	UCM: RXPVX = LEVEL + ERROR

Table 7.12 Model selection Private Insurance Payment: Cipro

We have also investigated the prediction error for the normal curve and kernel density estimation for private insurance payments. Figure 7.14 reveals that the normal curve fitting is closer to the kernel density estimation (for unknown distribution). The occurrence of the outlier on September, 2002 caused the gap between the two fits.

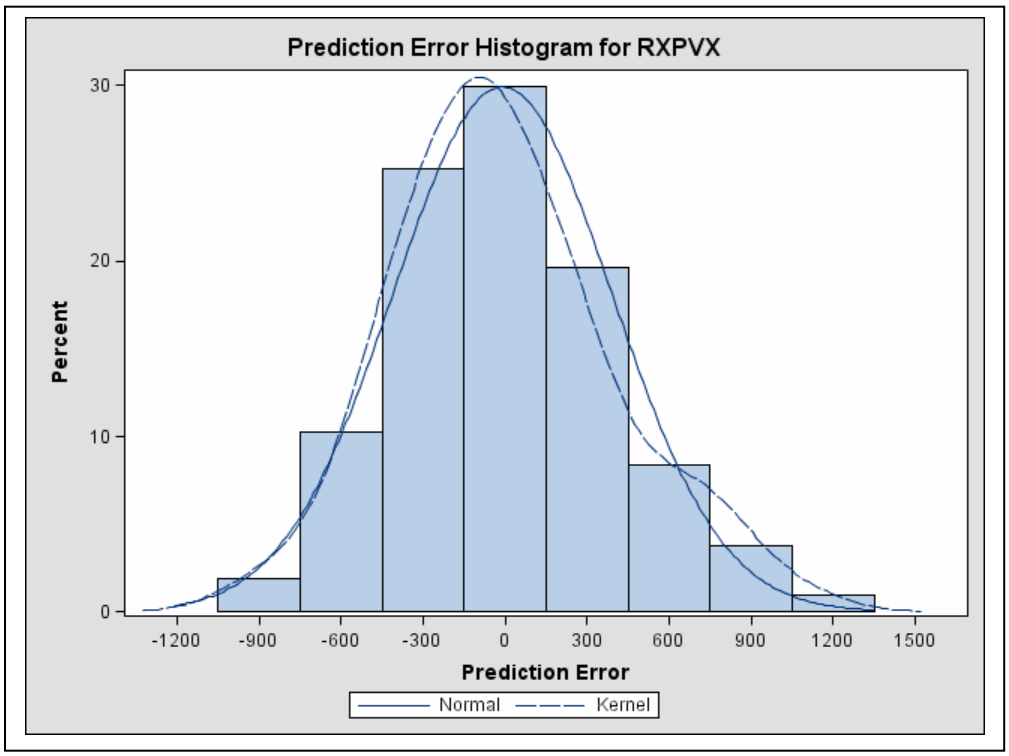

Figure 7.15 Prediction error for Insurance Payment

We have also investigated the prediction error for autocorrelation plot, partial autocorrelation plot and inverse autocorrelation plot. Each observation in a time series is correlated with previous prescriptions made, so it is important to analyze the autocorrelation function plots.

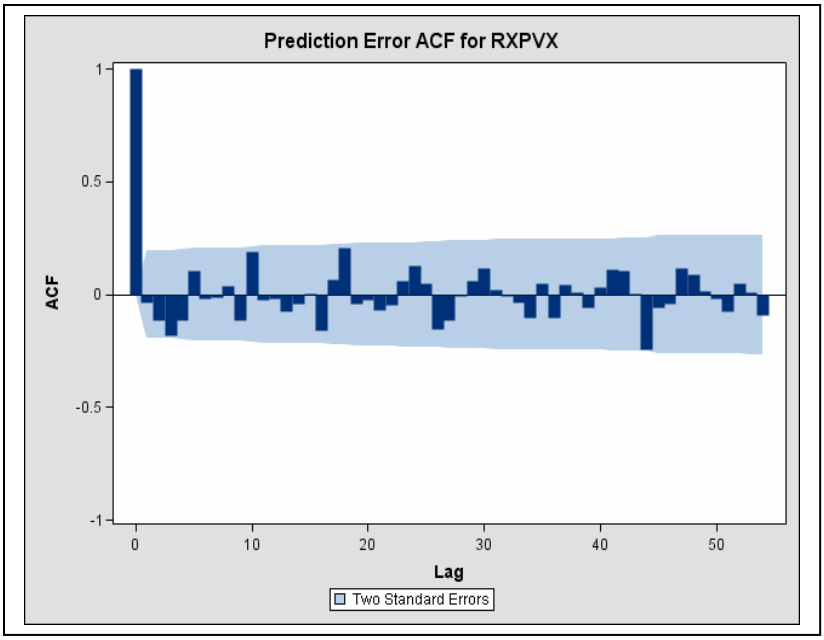

Figure 7.16. ACF plot of Private Insurance payment of Cipro

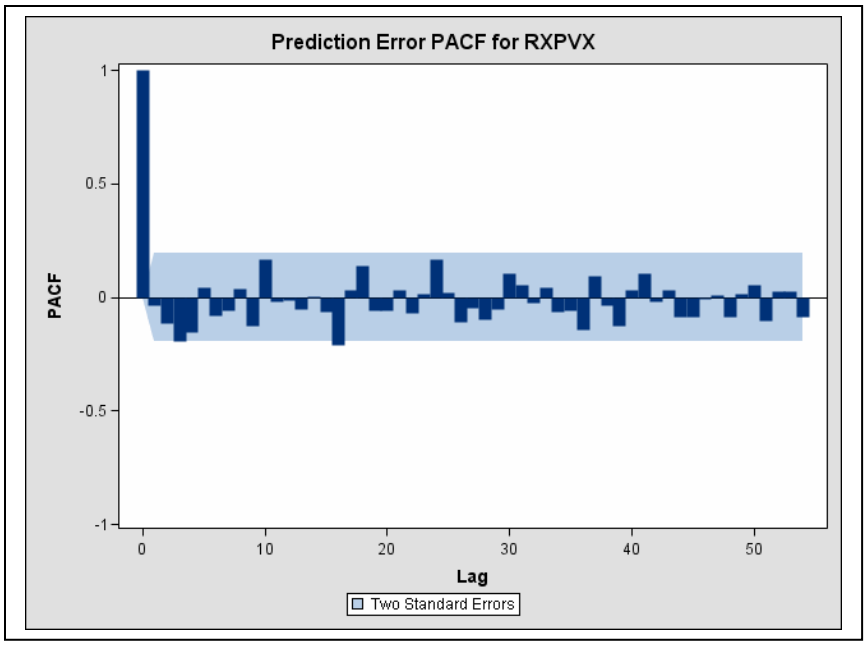

Figure 7.17 PACF plot of Private Insurance payment of Cipro

The error prediction for the autocorrelation function (figure 7.16) and partial autocorrelation function (figure 7.17) for the ARIMA model built shows that the correlations are within the bound of two standard errors; this reveals that the models are fitting the data well. As a consequence, we might adopt the ARIMA model as a model forecasting for the private insurance payment of the antibiotic, Cipro. The autocorrelation function plot and partial autocorrelation plot are also associated with the forecast model plot. The more the lags are outside the bounds of confidence, the more the model and forecast diverge. We also investigated the predictive model for private insurance payments for Cipro.

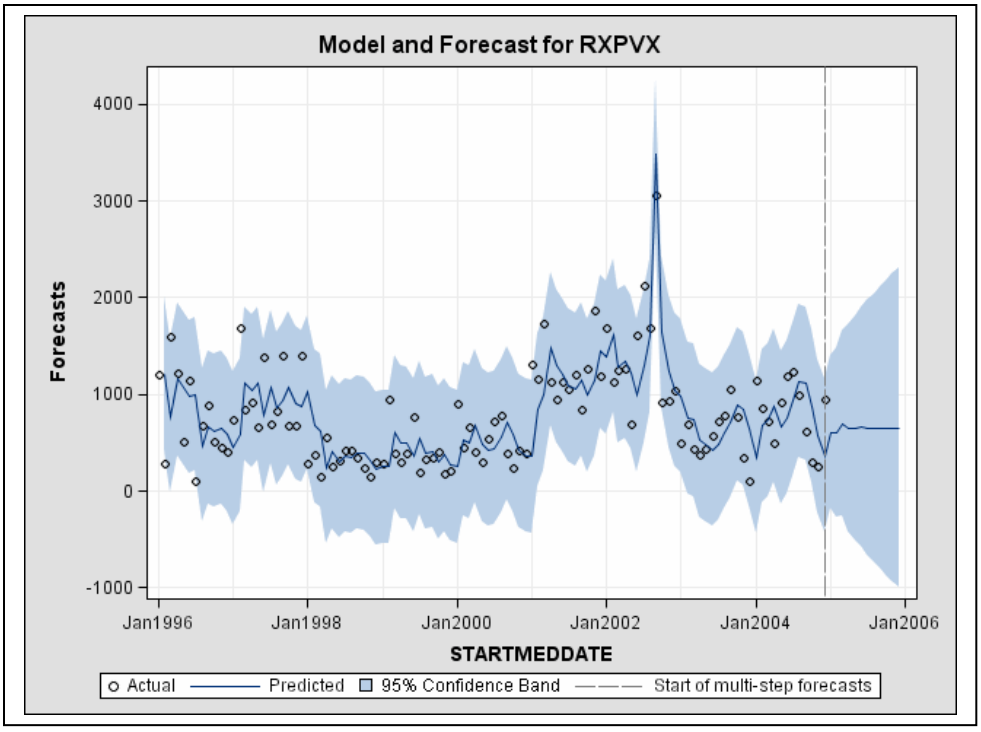

Figure 7.18 Model and Forecast for Private Insurance Payment of Cipro

The points in figure 7.18 represent the monthly private insurance payments for the antibiotic, Cipro and the thick blue line is the predictive model that we built. Most of the observations, with very few exceptions, lie within the 95% confidence band. There is a fairly constant increase in Private Insurance payments until the middle of January, 2001; then there is a sudden increase in September, 2002. As mentioned earlier, in time series forecasting, we use the historical data to predict or forecast what tomorrow's value will be based on today's value.

The fact of the matter is that we chose to forecast for two years (year 2005 and year 2006) to get a better fit of the data; otherwise, we could forecast for the next

50 years and so forth. But forecasting for a very long period of range gives a poorer fit; the reason is that there might be some other covariates that will occur in the near future that influence the forecast method used at the present time.

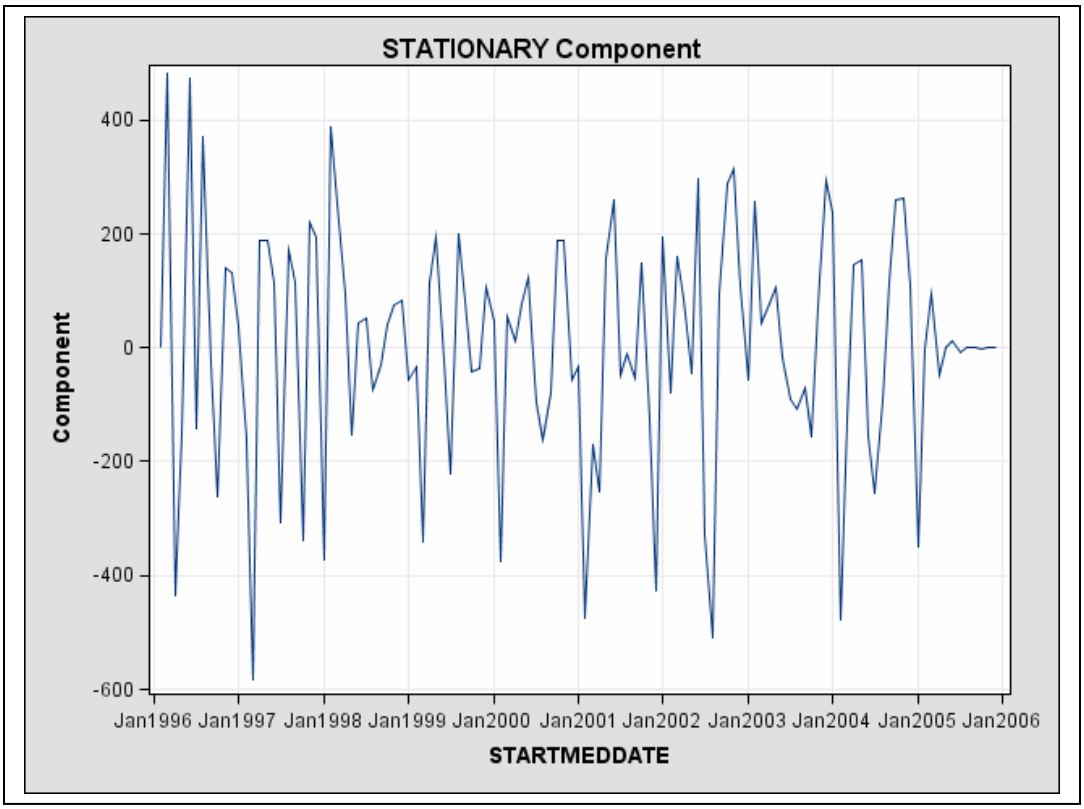

Figure 7.19 Stationarity Component Private Insurance for Cipro

Figure 7.19 is the stationary component of the model that we built, and we can see that after differencing the data, which was done by the ARIMA procedure given in SAS CODE 9, the series is stationary.

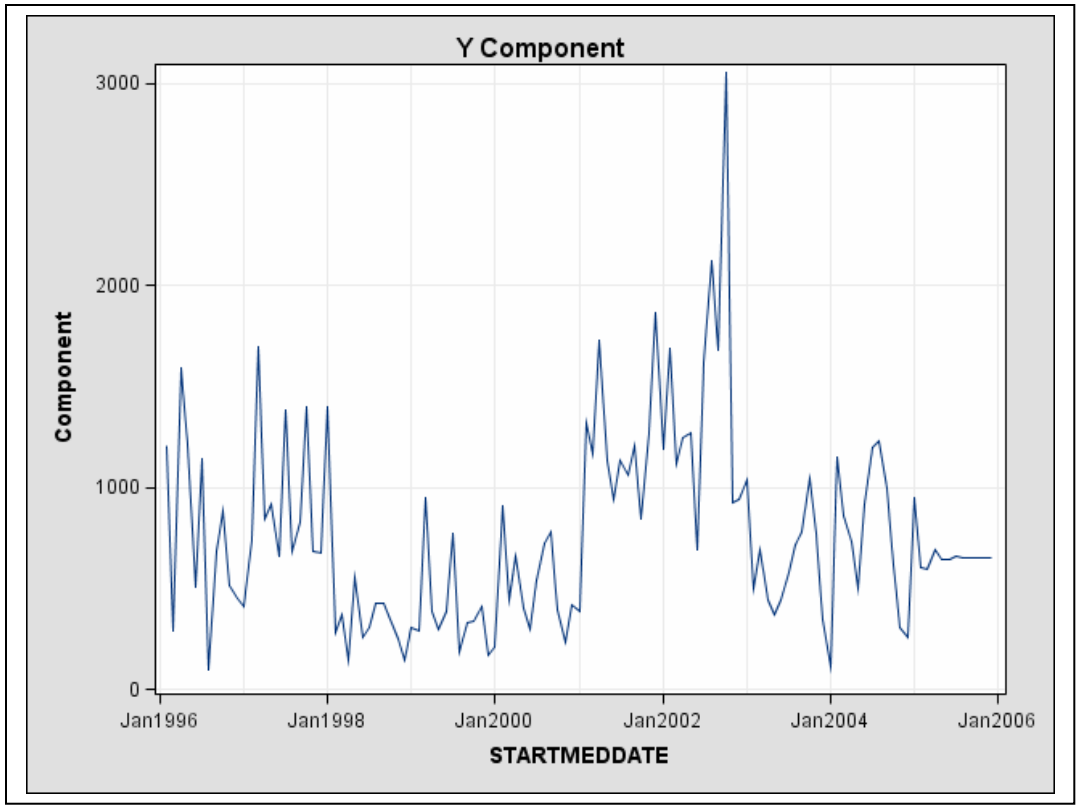

Figure 7.20 Y Component of Private Insurance for Cipro

Figure 7.20 is obtained by plotting each data point of the series forecast and connecting them by a line. This plot also reveals that in September, 2002, there is a sudden increase, which we define as an outlier. Figure 7.21 plots the outlier or event obtained through the model that we built. Outliers or events can be taken as single data points that we call point interventions in a time series. We will consider the data point of September, 2002 as a single data point; in other words, we will construct a dummy variable, and see if this can improve the forecast plot.

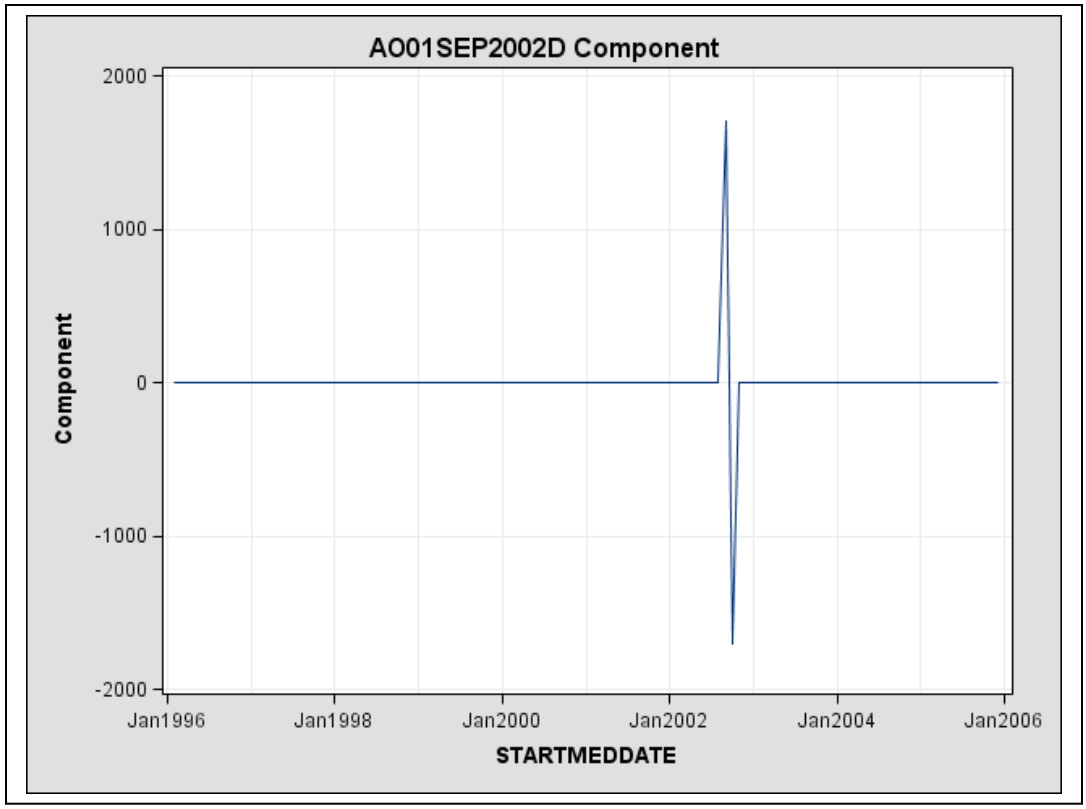

Figure 7.21 Outlier: Private Insurance for Cipro

The payments made for all the antibiotics in our dataset were calculated without taking inflation rate as a factor. But from an economic standpoint, inflation rate is important to consider when determining the price of items at a given period of time. As the inflation rate increases, the price of commodities increases as well and vice versa. Therefore, we introduced inflation rate as a dynamic regressor and compared the model error with the one without a dynamic regressor. The difference between regressor variable and dynamic regressor is that the latter uses past values of the predictor series, so that it will help us to model effects that take place gradually. We added inflation rate to the dataset of Amoxicillin

and investigated whether considering inflation rate as a dynamic regressor improves the model forecast.

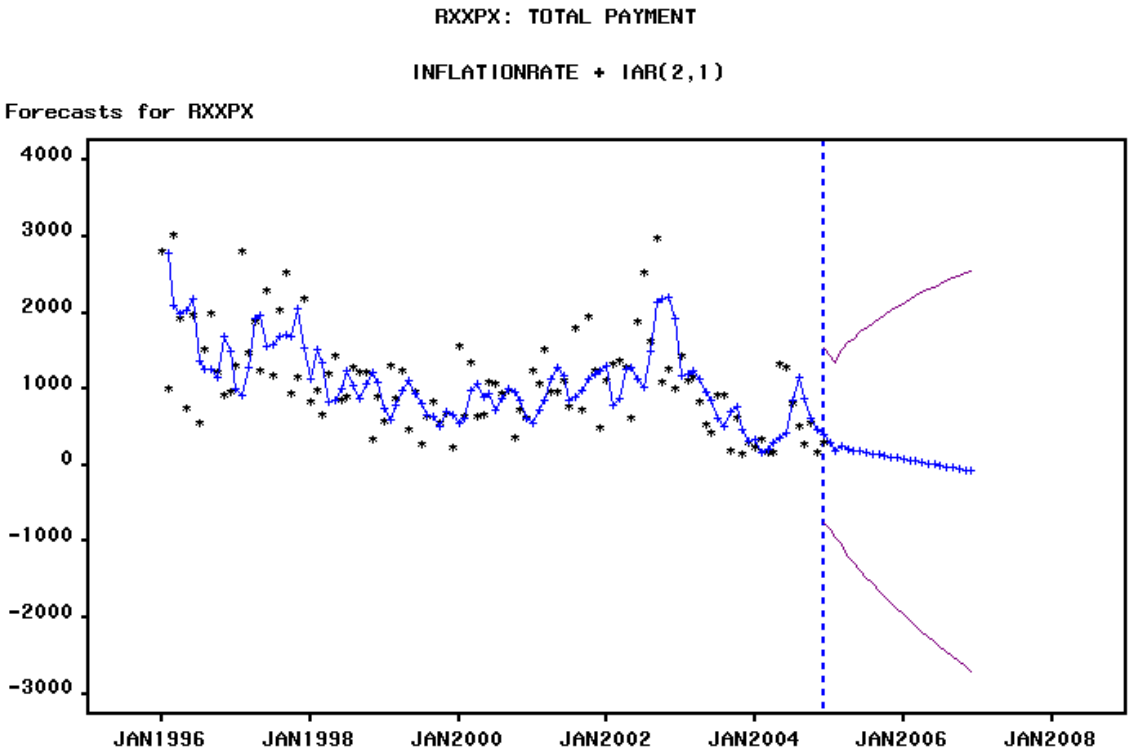

Figure 7.22: Forecasting Plot of Cipro using Inflation as Dynamic Regressor

The new model, which includes a dynamic regressor (figure 7.22), fits the data much better than ARIMA (2, 1, 0) without a dynamic regressor (figure 7.18). The predicted private insurance payment for the years 2005-2006 was constantly decreasing when a dynamic regressor was introduced. One possible reason is that the inflation rate affects the payments made for private insurance, and also the inflation rate was increasing through the time period. The root mean square

error was dropped from 559 to 430, which is an indication that introducing the dynamic regressor improved the model forecast. The plot of inflation rates from 1996-2004 is given in figure 7.23.

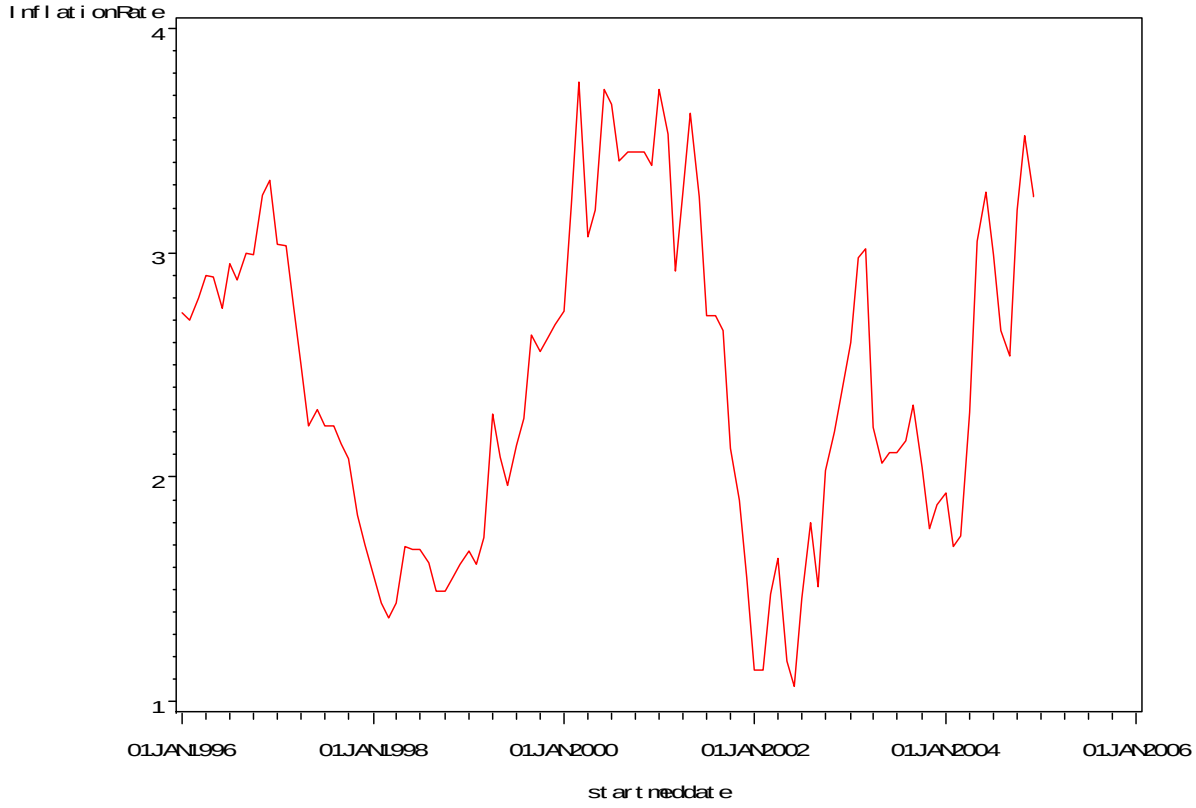

Figure 7.23: Inflation Rates as a Dynamic Regressor

Heteroskedasticity and GARCH Models

In chapter five, we talked about approaches to dealing with heteroskedasticity. The ordinary regression model is used when the errors have the same variance throughout the time points; in such a scenario, the data are called Homoscedastic. On the other hand, if the errors have non constant variances, then the data are called Heteroskedastic. Erroneously using ordinary least-squares regression for Heteroscedastic data causes the ordinary least squares estimate to be inefficient. As a consequence, we look for models that account for the changing variances that make efficient use of the data. On the other hand, heteroskedasticity can make the ordinary least squares estimates forecast error variance inaccurately, as the predicted forecast variance is based on the average variance instead of the variability at the end of the series [33].

The weighted regression method is a good method if the error variance at different time points is known, but if the error variance at different time points is not known, we must estimate it from the data by modeling the changing error variance. The generalized autoregressive conditional heteroskedasticity (GARCH) will be used to model for the heteroskedasticity errors. The analysis of the antibiotic, Erythromycin given in the previous section of this chapter reveals that there is higher variability in the beginning of the series that is given in figure 7.9. A GARCH model is a weighted average of past squared residuals that has a

declining weight on which recent observations are given larger weight than distant, past observations. Even if the series of time points given in figure 7.9 looks Heteroskedastic, we need to test it statistically.

```
Ods html;
Proc autoreg data=hpfertro;
 model RXXPX = STARTMEDDATE / nlag=12 archtest dwprob
noint;
     output out=out out=RXXPXresid;
run;
ods html close;
```

SAS CODE 11

Table 7.13 SAS code test heteroskedasticity

We used SAS CODE 11 to test for heteroskedasticity, by regressing Private Insurance payments on start mediation date and we used the ARCHTEST option to test for Heteroscedastic ordinary least squares residuals. We used the DWPROB option to test for autocorrelation.

Ordinary Least Squares Estimates			
SSE	4030688.02	DFE	107
MSE	37670	Root MSE	194.08756
SBC	1448.12303	AIC	1445.4409
Regress R-Square	0.4108	Total R-Square	0.4108
Durbin-Watson	0.3475		

Table 7.14 Ordinary Least Squares Estimates

Q and LM Tests for ARCH Disturbances				
Order	Q	Pr > Q	LM	Pr > LM
1	50.0071	<.0001	48.7292	<.0001
2	77.6249	<.0001	49.1592	<.0001
3	80.8958	<.0001	56.3419	<.0001
4	81.1530	<.0001	56.4903	<.0001
5	81.3417	<.0001	59.5441	<.0001
6	82.0714	<.0001	59.5462	<.0001
7	84.4500	<.0001	59.5559	<.0001
8	88.7266	<.0001	60.2213	<.0001
9	95.2285	<.0001	60.9780	<.0001
10	96.1345	<.0001	66.4472	<.0001
11	96.4317	<.0001	66.6586	<.0001
12	97.0906	<.0001	67.2222	<.0001

Table 7.15 Q and LM Tests for ARCH Disturbances

The Q statistic tests for changes in variance across time using lag windows ranging from 1 through 12. The *p*-values for the test statistics are significant and strongly indicate heteroskedasticity, with $p < 0.0001$ for all lag windows. The Lagrange multiplier (LM) tests also indicate heteroskedasticity. Both Q statistics and the Lagrange multiplier help to determine the order of the ARCH model needed for modeling the heteroskedasticity, on which the changing variance is assumed to follow an autoregressive conditional heteroskedasticity model.

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
STARTMEDDATE	1	0.0109	0.001261	8.64	<.0001

Table 7.16 Parameter Estimate

Ordinary Least Squares Estimates

The parameter estimates given in table 7.15 are also significant, an indication that the data have a Heteroskedastic property. Once we checked for heteroskedasticity of the data, then we used a generalized autoregressive conditional heteroskedasticity model (GARCH) that takes care of the

heteroskedasticity of the data. SAS CODE 12 was used to build a GARCH model and test the parameter estimates.

```
ods html;
Proc autoreg data=hpfertro;
model RXXPX = STARTMEDDATE / nlag=12 garch=(q=1,p=1) maxit=500
noint;
 output out=out cev=vhat;
Run;
ods html close;
```

SAS CODE 12

Table 7.17 SAS code to build GARCH model for Private Insurance of Cipro

SAS CODE 12 was used to build GARCH (1, 1), by going back 12 lags into the past. This code will test how many autoregressive orders are needed for the Heteroskedastic variable, Private insurance payments, made for the antibiotic, Erythromycin. We fitted an AR(12) and GARCH(1,1) model for the Private Insurance Payment series regressed on start antibiotic date. The AR(12) specifies an autoregressive error of order 12, while GARCH(1,1) specifies a conditional variance model. SAS CODE 12 will compute the estimated conditional error variance at each time period in the variable VHAT (estimated conditional error variance series) and output the dataset named OUT.

GARCH Estimates			
SSE	1263311.36	**Observations**	108
MSE	11697	**Uncond Var**	14945.1458
Log Likelihood	-655.97225	**Total R-Square**	0.8153
SBC	1382.17648	**AIC**	1341.94451
Normality Test	1918.0137	**Pr > ChiSq**	<.0001

Table 7.18 GARCH Estimates

The normality test is significant ($p < 0.0001$), which is consistent with the hypothesis that the residuals from the GARCH model, $\varepsilon_t / \sqrt{k_t}$, are normally distributed. The parameter estimate is significant. The parameter estimates given in table 7.19 include rows for the GARCH parameters. ARCH0 represents the estimate for the parameter ω, ARCH1 represents α_1, and GARCH1 represents δ_1. The parameter estimates for the autoregressive errors are significant up to lag 1; as a result, we adopt AR(1). Also, the GARCH1 parameter estimate is significant. The model for Erythromycin Private Insurance is therefore built with AR(1) + GARCH(1,1) on which the heteroskedasticity nature of the data is controlled. As we have discussed in the beginning of this chapter, an AR(2) was the best model predicted with a negative correlation of R^2=-0.627.

The autoregressive order of lag one is the only one significant, so we don't need lags of up to 12. The estimate of the mean term in the Hetereskedastic process is 0.0850, which is not significant, but the estimate of coefficient of the square of error terms 0.2103 is significant. The estimate of error variance at lag 1 is 0.7376, which is significant at a 5% level of significance.

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
STARTMEDDATE	1	0.0185	0.005313	3.48	0.0005
AR1	1	-0.5830	0.2065	-2.82	0.0047
AR2	1	-0.3273	0.2008	-1.63	0.1030
AR3	1	0.0545	0.2386	0.23	0.8195
AR4	1	0.1816	0.2391	0.76	0.4474
AR5	1	-0.0956	0.2075	-0.46	0.6448
AR6	1	-0.0242	0.2607	-0.09	0.9260
AR7	1	-0.1437	0.2584	-0.56	0.5782
AR8	1	0.0351	0.3917	0.09	0.9286
AR9	1	0.0115	0.3367	0.03	0.9726
AR10	1	0.0312	0.2624	0.12	0.9055
AR11	1	-0.0825	0.2040	-0.40	0.6858
AR12	1	-0.004789	0.1342	-0.04	0.9715
ARCH0	1	0.0850	0.0757	1.12	0.2614
ARCH1	1	0.2103	0.0847	2.48	0.0130
GARCH1	1	0.7376	0.0960	7.68	<.0001

Table 7.19 GARCH parameter estimates

We also investigated the average Medicare and Medicaid payments made for prescriptions of antibiotics. The term Medicaid is referred to as the amount of support in terms of healthcare given to low income people. A family is on the category of low income if the percentage of income of the household is less than

15,000 [35]. Medicaid only consists of a very small portion of the total payment made; that means the majority cost is paid by the patient. The government pays only a small portion of the total payment. We have built a model of prediction for both Medicare and Medicaid payments made for the antibiotic, Cephalexin. Cephalexin is a commonly prescribed antibiotic. A model of forecast was built for the Medicaid payment of Cephalexin; a double (brown) exponential smoothing model was built. The Medicaid payment was increasing through the years 1996 up to 2004, and the forecasted Medicaid payment for the years 2005 and 2006 was increasing as well; this means the government was paying more money for Medicaid than the previous years. One has to keep in mind that this increase in payment is due to cost increase; insurance payments increase overall due to total payment increase.

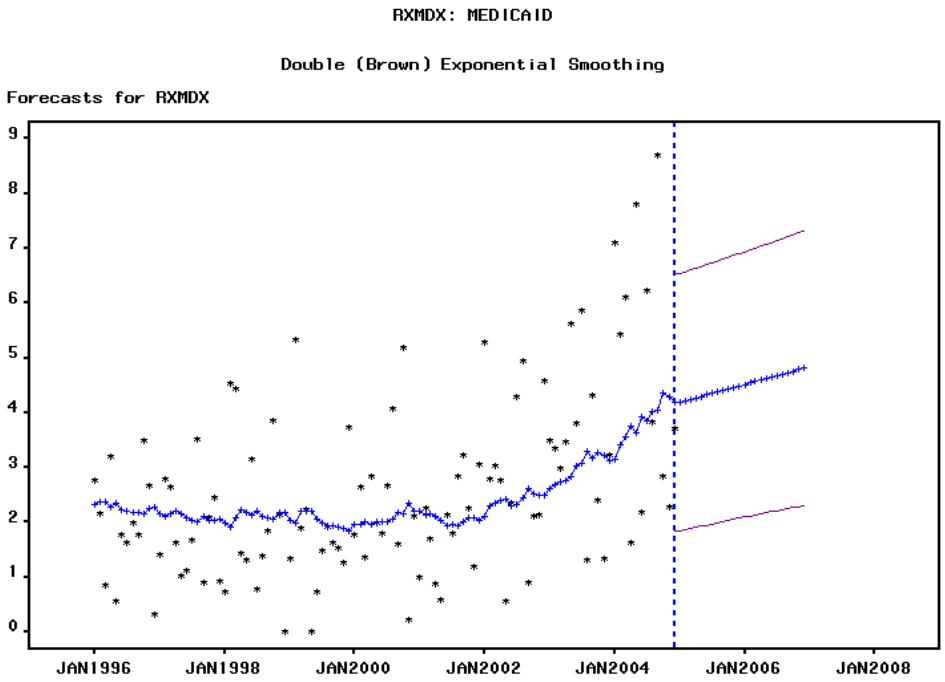

Figure 7.24 Model forecast for Medicaid payment: Cephalexin

We also investigated the average Medicare payments made for the prescription of Cephalexin. The term, Medicare, is referred to as the amount of support in terms of healthcare given to the American elderly people age of 65 or more. People with disabilities are also eligible for Medicare payments. Figure 7.25 indicates that on average, the government pays a maximum of less than three dollars for the prescription of Cephalexin. The data for Medicare payment has many missing observations, but in order to maintain the series of the data, we set missing values to zero; that is why a mean model was built as a model of prediction. As we can observe from figure 7.24, the model built did not capture most of the data; the reason being that the data has two observations of zero at November, 2004 and December, 2004. Statistically speaking, when we have as many observations missing, it is ideal to consider the mean as a model of prediction.

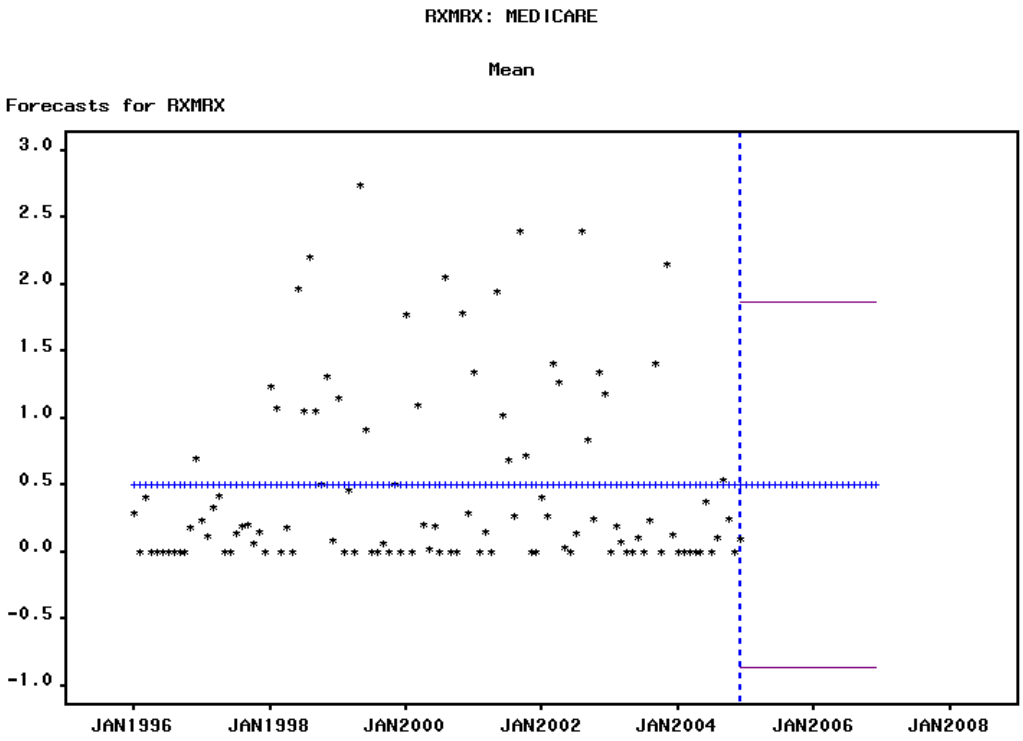

Figure 7.25 Model forecast for Medicare payment: Cephalexin

Comparing Medicaid payment (figure 7.24) and Medicare payment (figure 7.25), we see that Medicaid payments are three times as large as Medicare payments; further studies can be made if the number of low income people are three times as much as elderly or disabled people.

We finally built a model of forecast for the remaining antibiotic for the total payment made. As we have seen earlier in this chapter using the Means procedure, we found out that Amoxicillin was the most prescribed with 20,457 transactions while Cefuroxime was the least prescribed with only 28 transactions. The model forecast for all antibiotics was built using SAS CODE 13.

```
Proc sort data=diser.disertation;
by RXNAME STARTMEDDATE ;
RUN;
PROC HPF DATA=diser.disertation OUT=disertation LEAD=0;
ID STARTMEDDATE INTERVAL=MONTH ACCUMULATE=TOTAL;
FORECAST RXXPX RXPVX RXMDX RXMRX RXQUANTY DRUG
/MODEL=NONE;
BY RXNAME;
RUN;
Proc HPF data=disertation out=sasuser.forecast lead=24;
id startmeddate interval=month;
forecast RXXPX RXPVX RXMRX RXMDX RXQUANTY DRUG/select=mape
holdout=36;
BY RXNAME;
Run;                    SAS CODE 13
```

Table 7.20 SAS code for all antibiotics model building

Plotting the forecasted series of the antibiotics in one plot is the best way to compare the number of prescriptions, quantity of prescriptions and total payments made between several antibiotics. We plotted all twenty antibiotics in one graph, but as we can see from figure 7.26, we can hardly make comparisons; the reason being the difference in total number of transactions made for the antibiotics. As a result, we classified the antibiotics into classes of three based on the number of transactions.

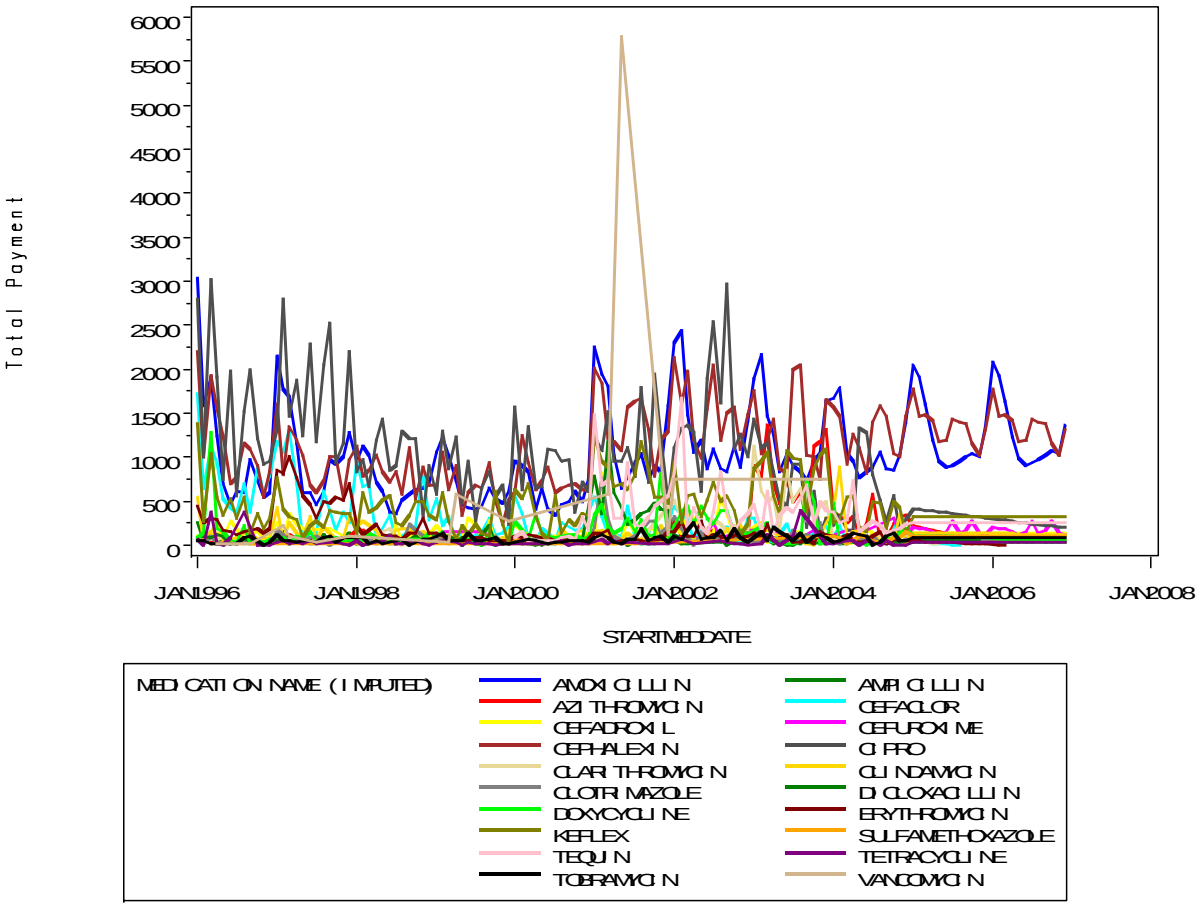

Figure 7.26. Model forecast for Total payments for all antibiotics

Figure 7.26 gives the forecasted plot of all the antibiotics for total payments made for all twenty antibiotics.

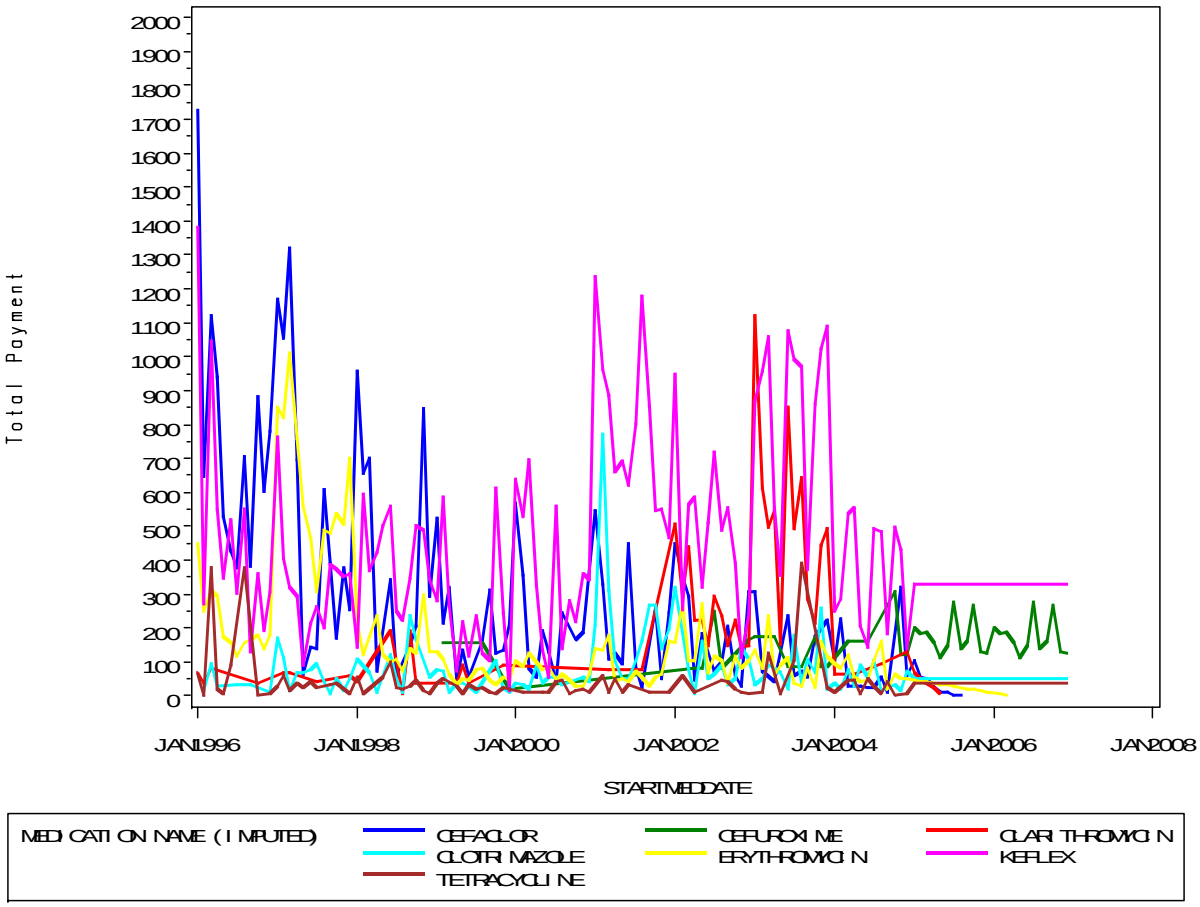

Figure 7.27. Model forecast for Cefaclor, Cefuroxime, Clarithromycin, Clotrimazole, Erythromycin, Keflex and Tetracycline: Total Payment

The total payments made for Keflex is rising starting in January, 2001, while the forecast series for Clarithromycin is decreasing.

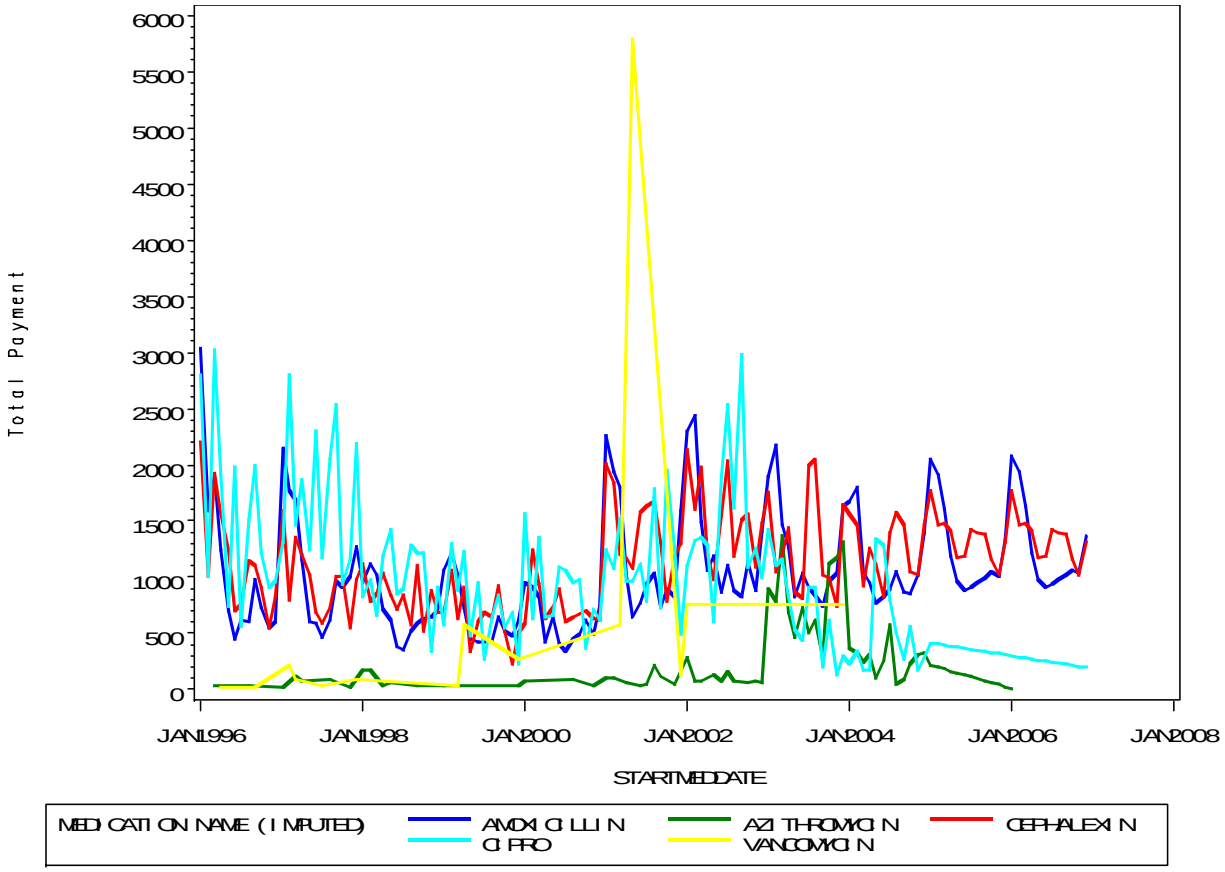

Figure 7.28. Model forecast for Amoxicillin, Azithromycin, Cephalexin, Cipro, and Vancomycin: Total Payment

The total payment for Vancomycin is highest on May, 2001; it is an outlier, while Ampicillin, Cipro, Azithromycin and Cephalexin increase through time with the forecast for Azithromycin decreasing.

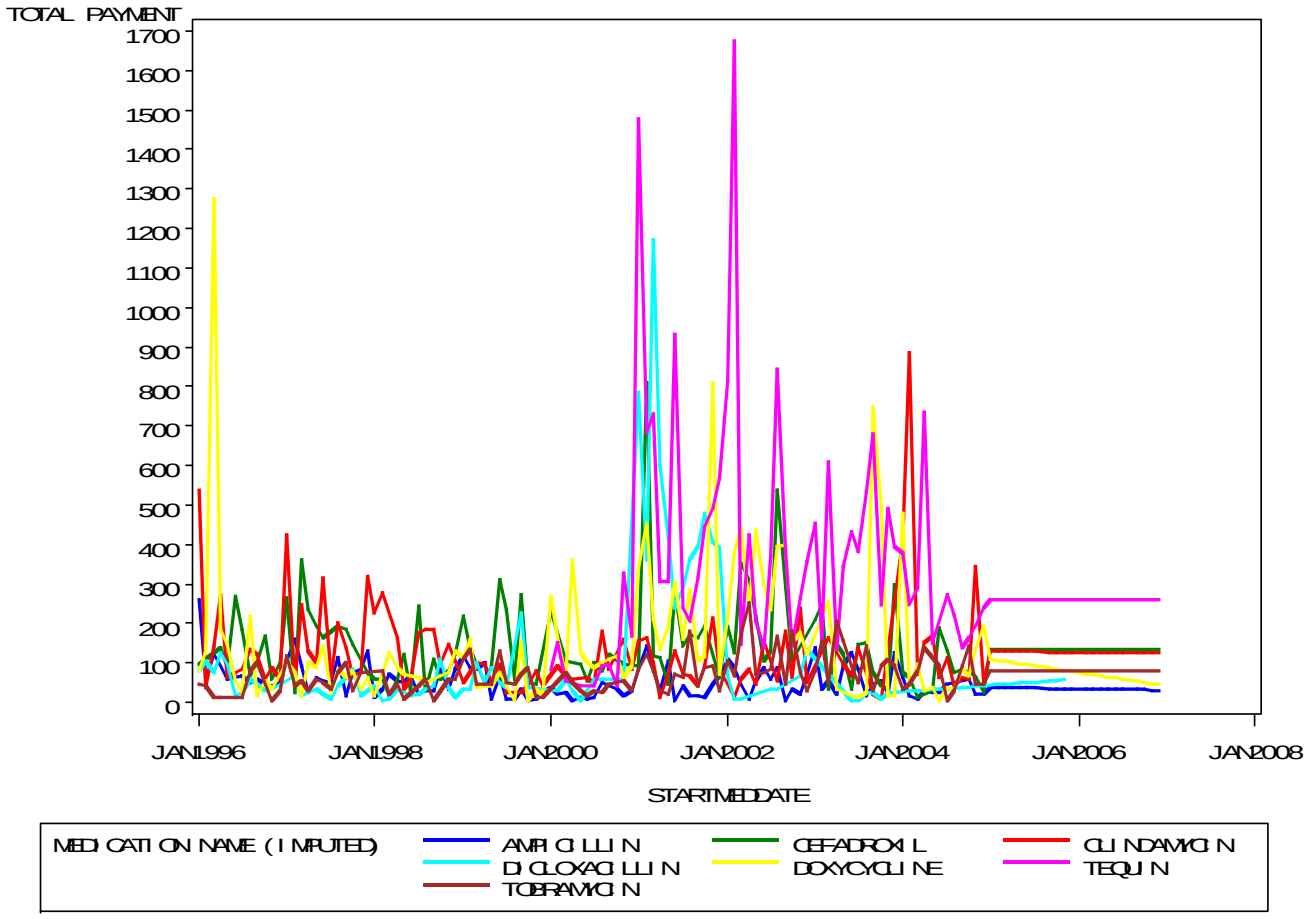

Figure 7.29. Model forecast for Ampicillin, Cefadroxil, Clindamycin, Dicloxacillin, Doxycycline, Tequin and Tobramycin: Total Payment

The total payment for Tequin is highest in January, 2002, while Dicloxacillin is at its peak around January, 2001.

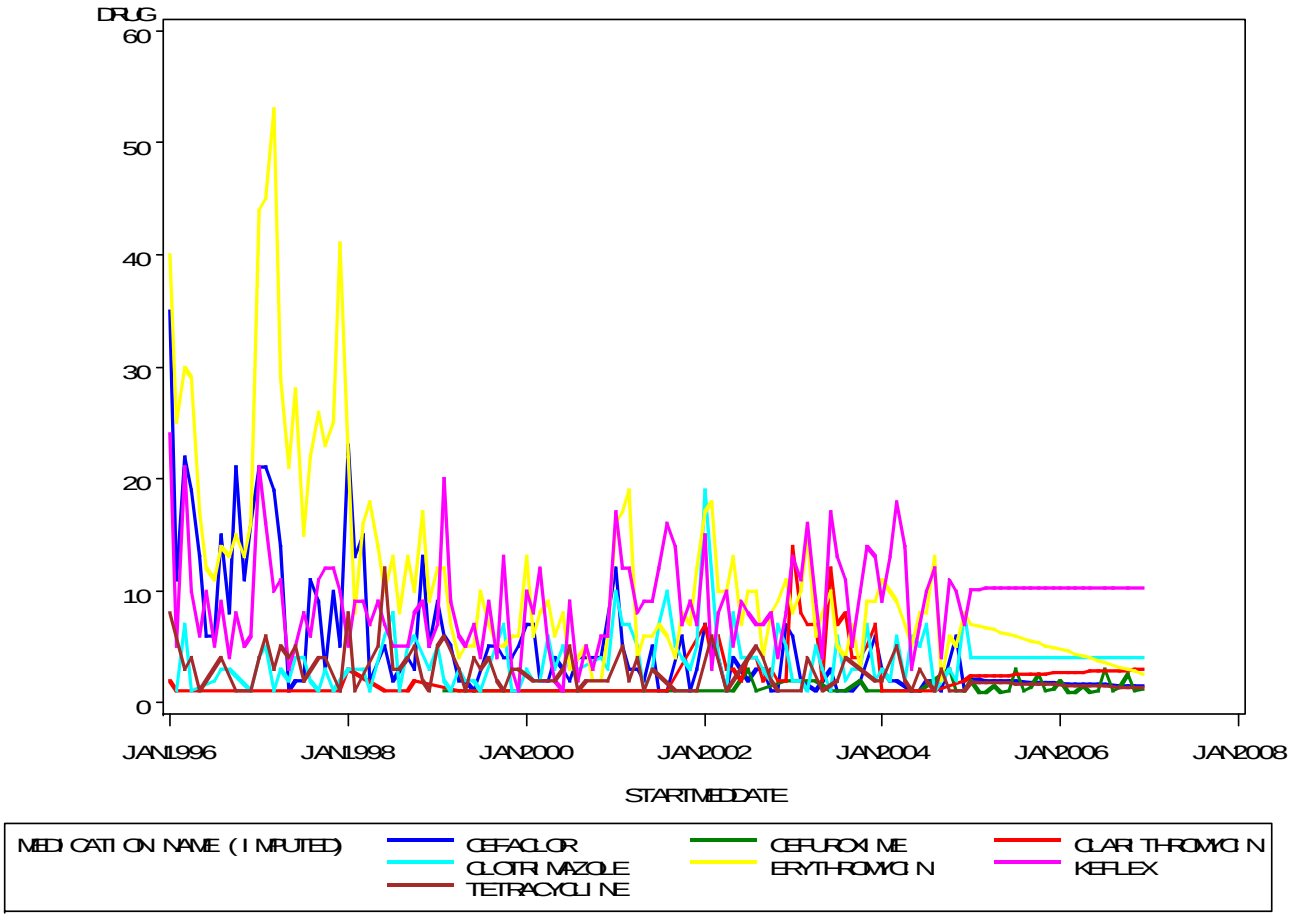

Figure 7.30. Model forecast for Cefaclor, Cefuroxime, Clarithromycin, Clotrimazole, Erythromycin, Keflex and Tetracycline: Number of Prescription

The number of prescriptions of Erythromycin was higher until January, 1999, and then the number of Keflex prescriptions surpassed the number of those for Erythromycin.

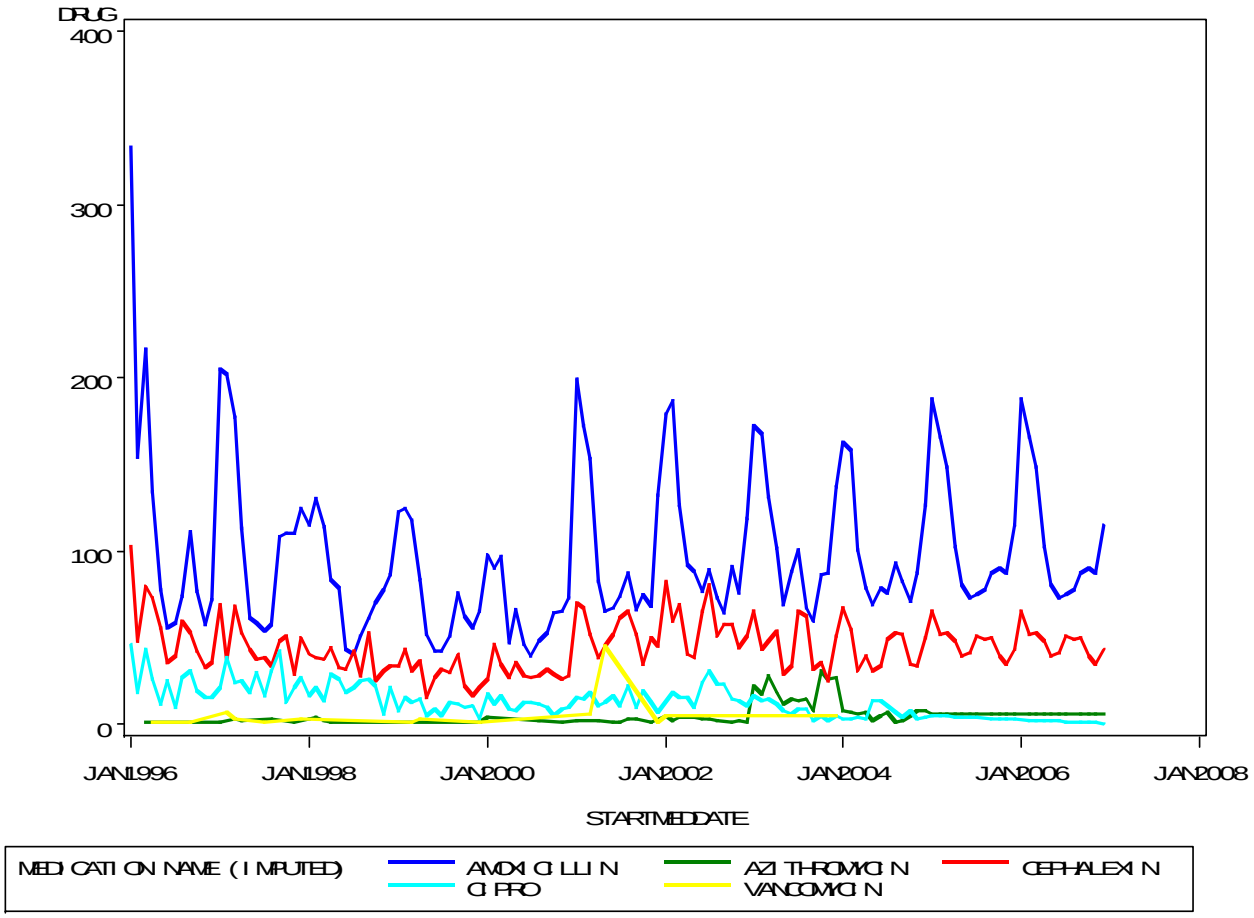

Figure 7.31. Model forecast for Amoxicillin, Azithromycin, Cephalexin, Cipro, and Vancomycin: Number of Prescriptions

The number of prescriptions of Amoxicillin was greater than the number for Azithromycin, Cephalexin, Cipro, Vancomycin, but all prescriptions were seasonally increasing through time.

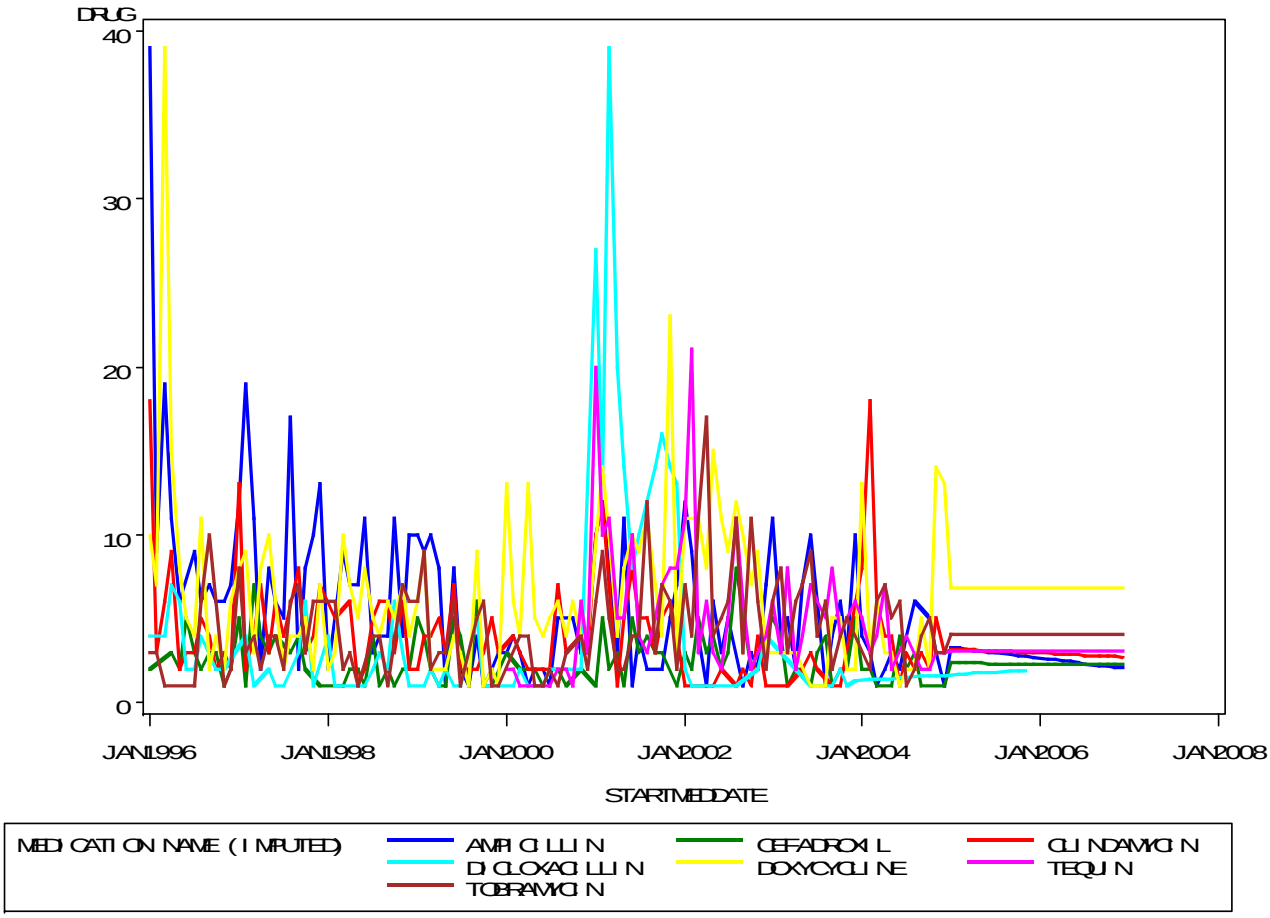

Figure 7.32. Model forecast for Ampicillin, Cefadroxil, Clindamycin, Dicloxacillin, Doxycycline, Tequin and Tobramycin: Number of Prescriptions

The number of prescriptions for Ampicillin, Cefadroxil, Clindamycin, Dicloxacillin, Doxycycline, Tequin and Tobramycin are close to each other, with Dicloxacillin at its highest value on January, 2001.

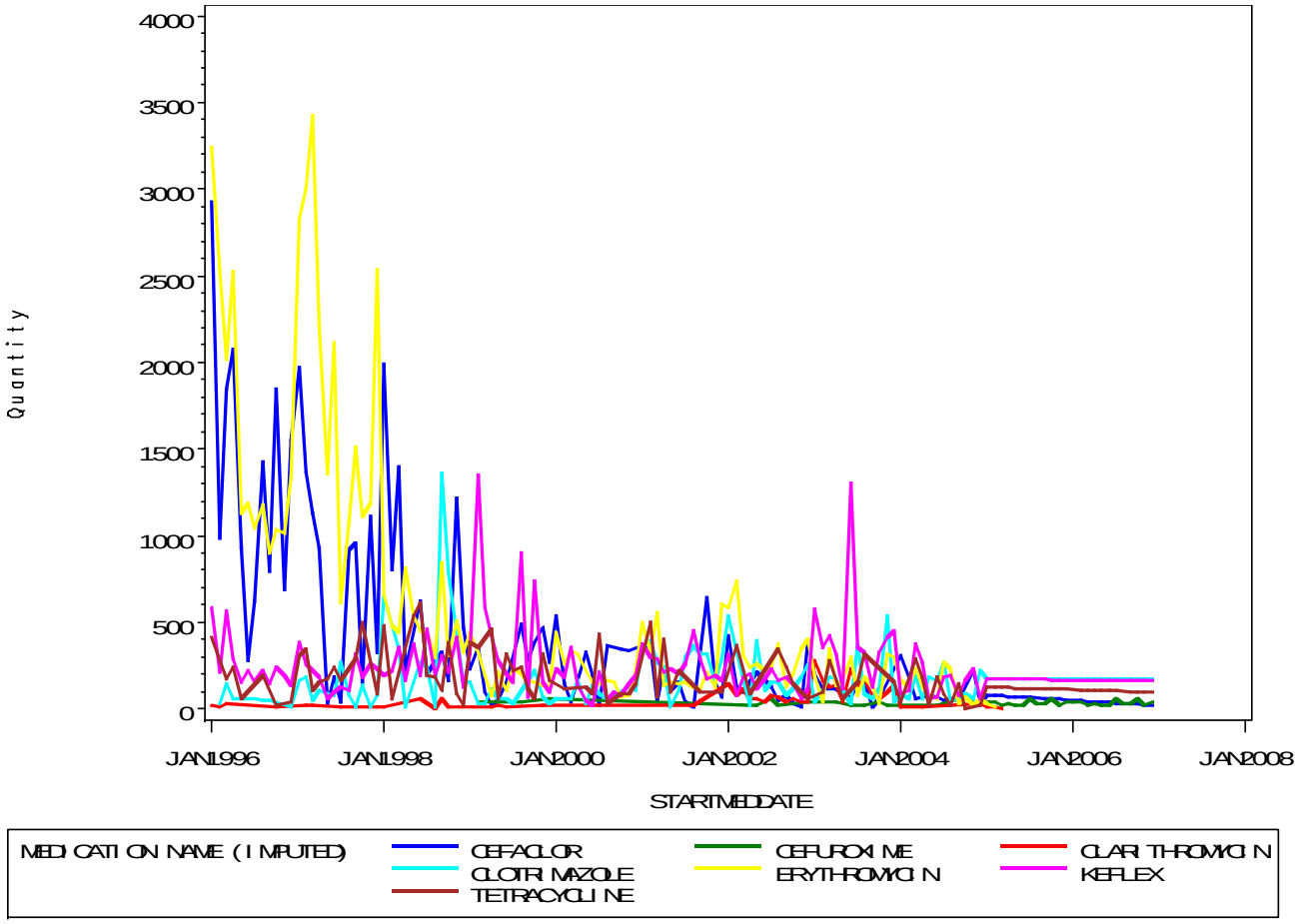

Figure 7.33. Model forecast for Cefaclor, Cefuroxime, Clarithromycin, Clotrimazole, Erythromycin, Keflex and Tetracycline: Quantity

The quantity of antibiotics prescribed is close for each antibiotic starting January, 1999 with a slightly higher quantity of prescriptions for Keflex around May, 2003.

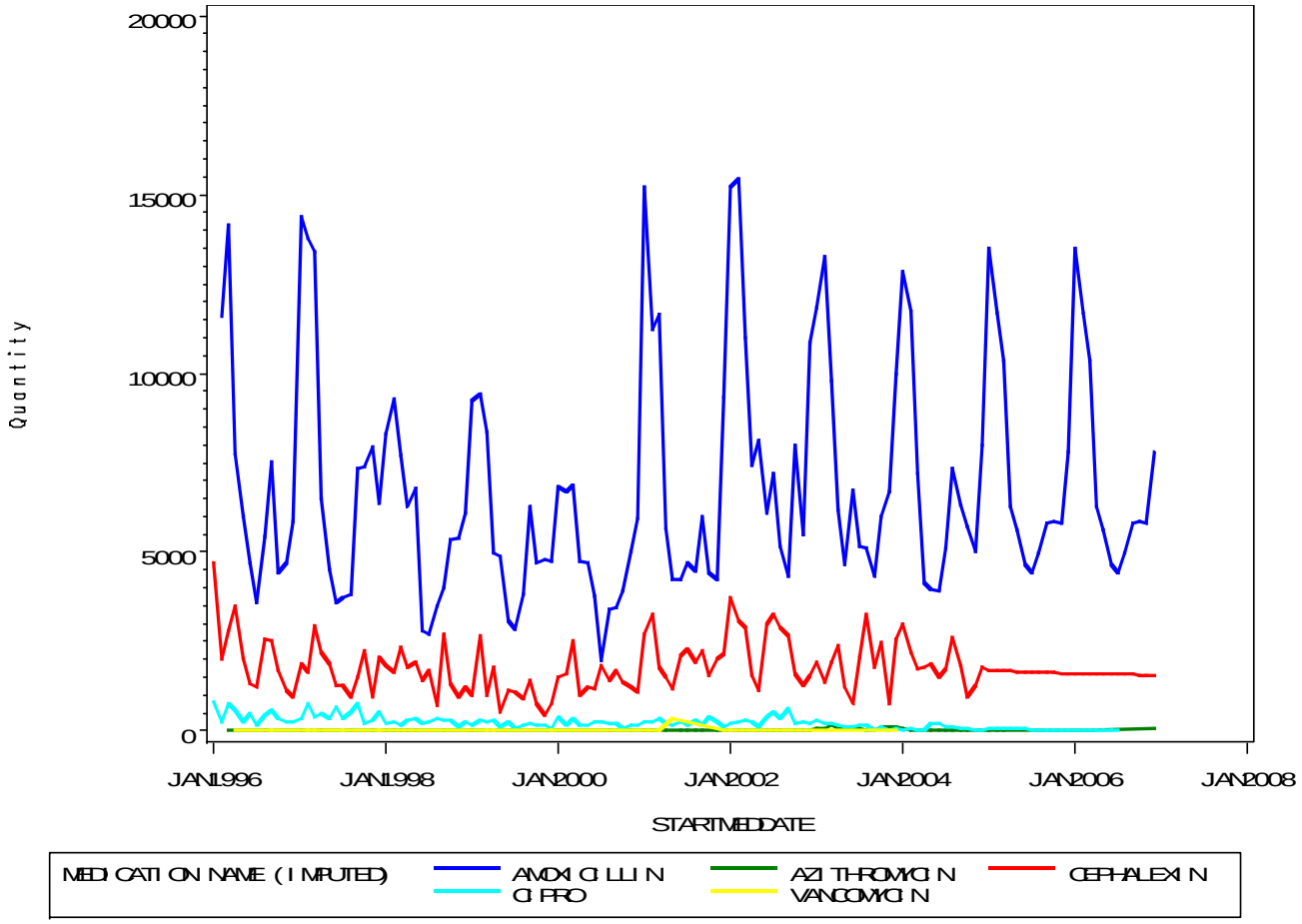

Figure 7.34. Model forecast for Amoxicillin, Azithromycin, Cephalexin, Cipro, and Vancomycin: Quantity

Amoxicillin prescription quantity is higher compared to Cipro, Azithromycin, Cephalexin and Vancomycin, while Cipro and Vancomycin have the smallest quantity of prescriptions.

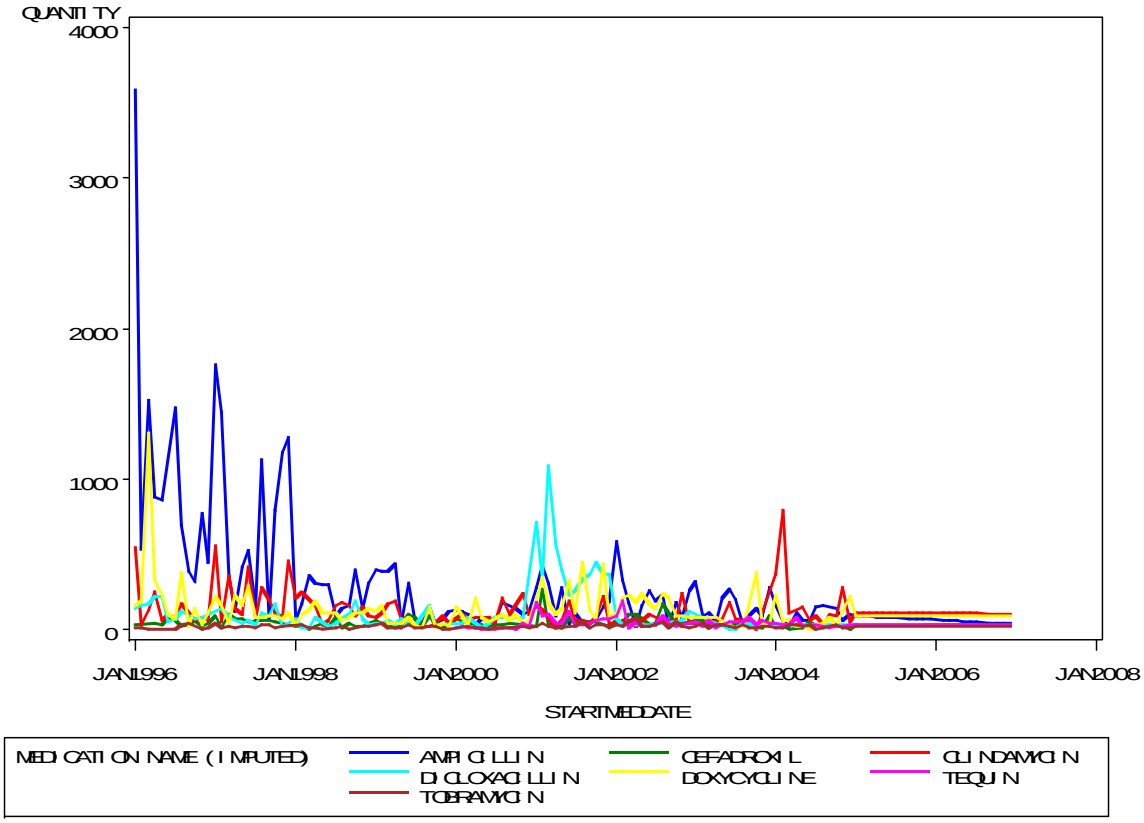

Figure 7.35. Model forecast for Ampicillin, Cefadroxil, Clindamycin, Dicloxacillin, Doxycycline, Tequin and Tobramycin: Quantity

Ampicillin prescription quantity is higher from January, 1996 up to May, 1999, but the series of prescription quantity is about the same starting in January, 2000, with a sudden peak for Dicloxacillin, Ampicillin and Clindamycin.

From an economic stand point, we know that the cost of an item is determined by the market; that means that the supply and demand play a major part. By the same token, the cost of antibiotics depends on the number of prescriptions and the quantity of antibiotics sold. One has to test statistically whether the number of prescriptions and the quantity of prescriptions significantly predict the total payments made for the antibiotics. We regressed the total payment of antibiotic

on the quantity and the number of prescriptions. We tested statistically if the quantity and the number of antibiotics affects the total payments made on antibiotic. We picked the antibiotic, Amoxicillin, to demonstrate the effect of predictors (quantity and number of prescriptions) on predicting the total payment.

```
DATA AMOXICILLIN;
SET DISER.DISERTATION;
WHERE RXNAME IN ('AMOXICILLIN');
RUN;
PROC HPF DATA=AMOXICILLIN OUT=AMOXICILLIN LEAD=24;
ID STARTMEDDATE INTERVAL=MONTH ACCUMULATE=TOTAL;
FORECAST RXXPX/MODEL=NONE;
FORECAST RXQUANTY/MODEL=BESTS SELECT=MAPE;
FORECAST DRUG/MODEL=WINTERS TRANSFORM=LOG;
RUN;
PROC AUTOREG DATA=AMOXICILLIN;
        MODEL RXXPX=RXQUANTY DRUG;
        OUTPUT OUT=TOTAL P=PREDICTED;
LABEL RXQUANTY='QUANTITY';
RUN;                    SAS CODE 14
```

Table 7.21 SAS code for regression procedure of total payment of Amoxicillin

SAS CODE 14 was used to create the dataset, Amoxicillin, using the Data statement, with no predictive model for total payment, best seasonal model for

quantity of antibiotics, and Winter's model with a log transformation for the number of prescriptions (drug).

The AUTOREG Procedure

Dependent Variable RXXPX
TOTAL PAYMENT

Ordinary Least Squares Estimates

SSE	2513834.41	DFE	105
MSE	23941	Root MSE	154.72970
SBC	1406.49748	AIC	1398.45109
Regress R-Square	0.9117	Total R-Square	0.9117
Durbin-Watson	0.7658		

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Variable Label
Intercept	1	0.9126	34.2248	0.03	0.9788	
RXQUANTY	1	0.0445	0.0167	2.66	0.0091	QUANTITY
DRUG	1	7.1520	1.2771	5.60	<.0001	NUMBER OF PRESCRIPTIONS

Table 7.22 Parameter Estimates for Predicting Total Payment

The parameter estimates for RXQUANTY and DRUG are significant with a p-value of 0.0091 and <0.0001 at 5%. The significant parameter estimates the number of prescriptions and the quantity of antibiotics indicates that both can forecast the total payment made for Amoxicillin. About 91% of the time, the variation in total payment is explained by both the quantity and the number of antibiotics. The R-square value close to one indicates that both the predictor variables, quantity and number of prescriptions, predict the total payment.

Hence, the model prediction for total payment can include the quantity of antibiotic and the number of prescriptions. We also plotted the model forecast for total payment using the quantity of antibiotic and the number of prescriptions as predictors.

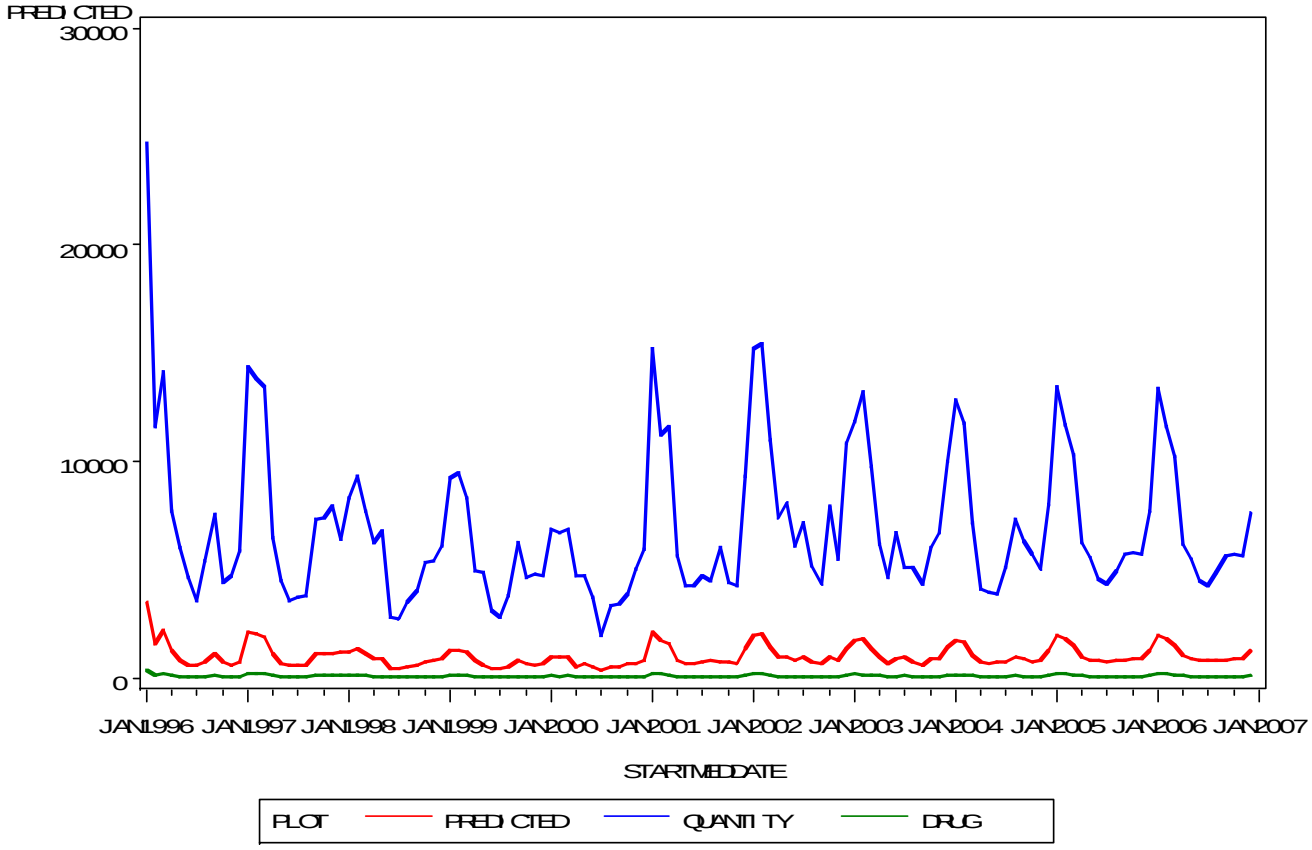

Figure 7.36. Prediction of Total Payment using Quantity and Number of Prescriptions for Amoxicillin.

When we introduced the quantity of prescriptions and the number of prescriptions as predictor variables for total payment for the prescription of Amoxicillin, the R^2 value dramatically increased from -0.094 to 0.9117; this is a great improvement of the predictor variables' influence on the total payment of Amoxicillin. Also,

when compared to total payments made for Amoxicillin without the predictor variable (figure 7.8), figure 7.36 gave a better forecast plot with the total payment seasonally increasing or decreasing as we move from January, 1996 up to December, 2006.

Patient Condition Severity

We then used text mining with kernel density estimation to reduce a large number of patient condition codes to make a comparison between the severity of the patient condition on the use of antibiotics. We condensed thousands of patient conditions into an index of 6 levels, and those levels are used to examine the relationship to different target variables of total payment and private insurance payments. Since there are hundreds of ICD-9 codes, and most patients have more than one ICD-9 code assigned to them, we compress the codes into a total of six clusters. We applied clustering and text mining to group similar patients together so that meaningful analyses can be performed examine cost. We used text mining to process and analyze the ICD-9 codes and to find similarities between patients. We then group similar patients together.

Clustering was performed using the expectation maximization algorithm. It is a relatively new, iterative clustering technique that works well with nominal data in comparison to the K-means and hierarchical methods that are more commonly

used. Six clusters were formed using the ICD9 codes based on the similarity of the text of each ICD-9 code.

```
Proc sort data = sasuser.Antibiotics out= work.sort_out;
    by duid rxicd1x;
Run;

options obs=max;

Data work.sort_out1;
    set work.sort_out;
          icd9 = translate(left(trim(rxicd1x)),'_',' ');
Run;
proc Transpose data=work.sort_out1 out=work.tran
                prefix=icd9_;
    var icd9 ;
    by duid;
run;
data work.concat( keep= duid icd9 ) ;
          length icd9 $32767 ;
          set work.tran ;
          array rxconcat {*} icd9_: ;
          icd9 = left( trim( icd_1 )) ;
          do i = 2 to dim( rxconcat ) ;
          icd9 = left(trim(icd9)) || ' ' || left(trim( rxconcat[i] )) ;
          end ;
run ;                                SAS CODE 15
```

Table 7.23 SAS code used to change ICD-9 codes to text and create clusters

```
Proc sql ;
        select max( length( icd9 )) into :icd9_LEN from work.concat ;
quit ;
%put icd9_LEN=&icd9_LEN ;
Data work.concat1 ;
        length icd9 $ &icd9_LEN ;
        set work.concat ;
Run ;
Proc contents data=work.concat1 ; Run;  SAS CODE 15
```

Table 7.23 SAS code used to change ICD-9 codes to text and create clusters (continued)

Using SAS CODE 15, six clusters were formed based on the ICD9 codes using Enterprise Miner 5.2. We created six clusters with the cluster number given in table 7.24. We will use these clusters to compare the distribution of antibiotics between the clusters using kernel density estimation. Table 7.25 is the description of the clusters shown in table 7.24.

#	Descriptive terms	Freq	Percentage
1	601,562,786,487,465,596,522,496,892,785,575,996,784	760	0.79249218
2	716	2	0.00208551
3	599,593,382,388,493,595	38	0.03962461
4	473,490,519,401,429,477,592,491,311,686,919,428,492, 486,595,478,590	108	0.11261731
5	780,v68,460,41,v25,272,	19	0.0198123
6	244,518,998,590,493,486	32	0.03336809

Table 7.24 Clusters of the ICD-9 Codes

Cluster Number	ICD-9 Codes	*ICD-9 Risk Factors*	Frequency	Label
1	601 562 786 487 465 596 522 496 892 785 575 996 784	Prostatitis Diverticula of Intestine Respiratory & Chest symptom Influenza Upper respiratory infection acute Bladder disorder Pulp disease & peripheral tissues Chronic airway obstruction Open wound of foot Symptoms involving cardiovascular system Disorder of gallbladder Anastomosis, graft (bypass), implant Symptoms of head and neck	760	Routine problems
2	716	Other and unspecified arthropathies	2	Arthritis
3	599 593 382 388 493 595	Other disorders of urethra and urinary tract Other disorders of kidney and ureter Suppurative and unspecified otitis media Other disorders of ear Asthma Cystitis	38	Urinary tract infection, asthma
4	473 490 519 401 429 477 592 491 311 686 919 428 492 486 595 478 590	Chronic sinusitis Bronchitis, not specifies as acute or chronic Other diseases of respiratory system Essential hypertension Complications of heart disease Allergic rhinitis, hay fever spasmodic rhinorrhea Calculus of kidney and ureter Chronic bronchitis Depressive disorder Local infections of skin and subcutaneous tissue Superficial injury Heart failure Chronic obstructive pulmonary disease Pneumonia, organism unspecified Cystitis, other disease of urinary system Other disease of upper respiratory tract Infections of kidney	108	Severe complications of respiratory system
5	780 v68 460 041 v25 272	Alteration of consciousness, hallucinations Persons encountering health service Acute nasopharyngitis (common cold) Bacterial infection Contraceptive management, sterilization Disorders of lipoid metabolism	19	Mild risk factors
6	244 518 998 590 493 486	Acquired hypothyroidism, like post-surgical Diseases of lung, pulmonary collapse Postoperative shock, hemorrhage Infections of kidney Asthma Pneumonia	32	Moderate risk factor

Table 7.25. Text Clusters Defined by Expectation Maximization

We used kernel density estimation to examine differences within the six clusters. Figures 7.24 and 7.25 are showing the graphs of total payments and private insurance payments respectively by cluster id for the antibiotic, Cipro. Note that cluster 2 has a high probability of total charges and reimbursements compared to the other clusters where the amount is very low. These graphs demonstrate a natural ordering in the clusters that is defined within the text mining tool.

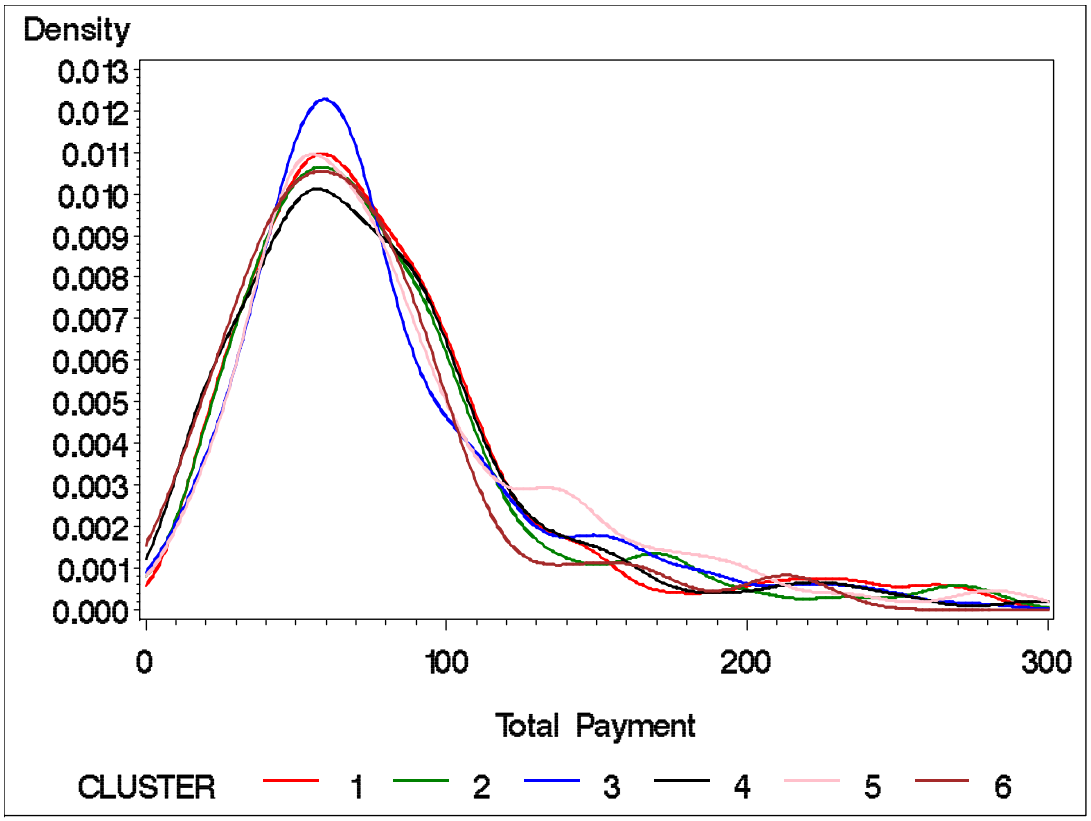

Figure 7.37 Distributions of Total Payments for CIPRO

The clusters formed based on the severity of the patient condition were compared using Kernel density estimation to examine total payments for the

antibiotic, Cipro (Figure 7.37). For instance, for patients in cluster 3 taking Cipro, they have a higher probability of paying between 40-80 dollars. As the total payment exceeds 280 dollars, the severity of the disease does not play a role on the size of payment made.

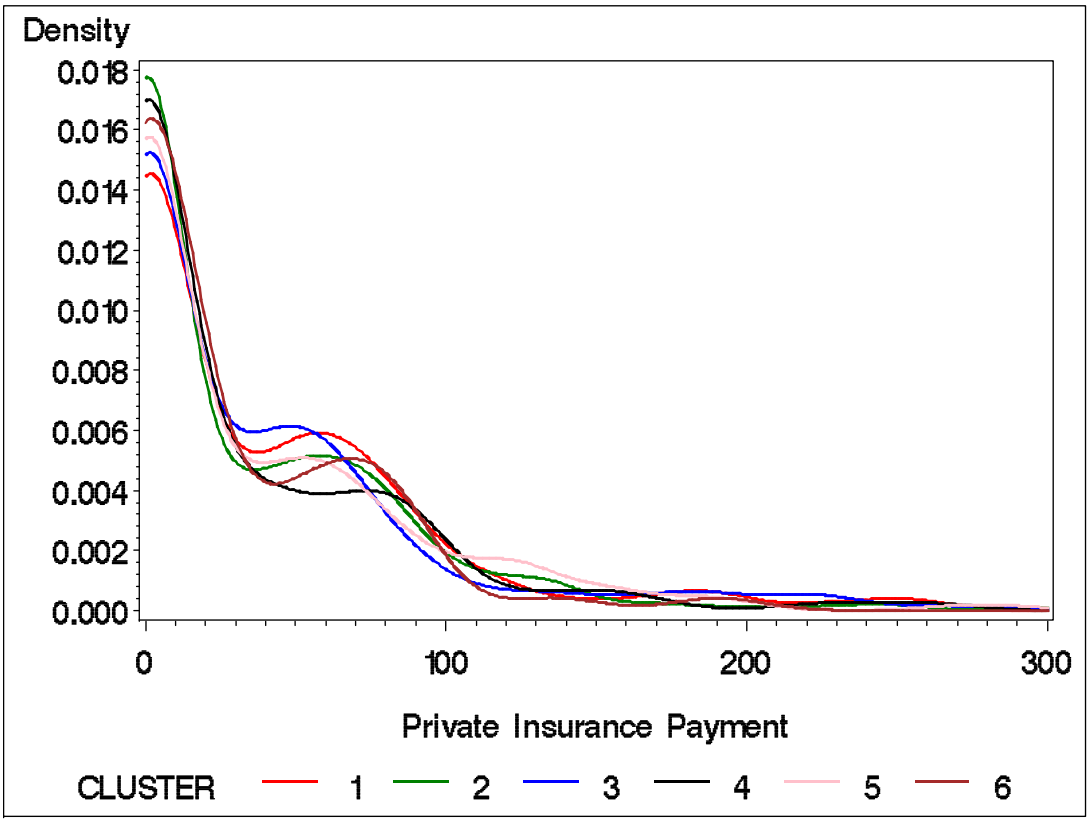

Figure 7.38 Distributions of Private Insurance Payments for CIPRO

The clusters formed based on the severity of the patient condition were compared using Kernel density estimation to examine private insurance payments for each antibiotic (Figure 7.38). For instance, for patients in cluster 3 taking Cipro, they have a higher probability of paying between 30-60 dollars but patients in cluster 6 have a higher probability of paying between 115-180 dollars.

Discussion

The rising cost of antibiotics is an important concern to health care providers and to society. For many years, proper application of antibiotics has been difficult to regulate and to control. Antibiotic costs have increased dramatically over the years with an overall trend to prescribe expensive, broad spectrum antibiotics rather than narrow-spectrum antibiotics. The term trend of antibiotics cost is important knowledge to health care providers and insurance companies. The trend analysis depends largely on an initial explanatory analysis of the data, and in identifying the appropriate models to predict the trend. The best approach currently available to model trend is to eliminate the trend by differencing and data correction and to find an appropriate stationary model for the differenced series.

Trend modeling requires finding the appropriate order of differencing, to correct the data for missing values, and to identify the appropriate order of stationary models for the differenced and corrected data [32]. We have investigated the changing behavior of the cost of antibiotics by introducing intervention variables. Usually, the increase of cost is related to a certain event; for instance, when September 11 happened, the airline industry lost tremendous amounts of business as a consequence of people not traveling by air.

In this dissertation, we are concerned not only with studying the increase in cost of antibiotics, but also with a sudden shift (both increase and decrease) in the cost of antibiotics. For instance, if we examine the private insurance payment made for the antibiotic, Cipro, we observe that there was a constant increase until a sudden and tremendous increase in September, 2002. Due to this tremendous increase, the forecast for the future is influenced if we don't take this spike into consideration. As a result, we introduced point and step interventions. A point intervention is assigned a value of one for September, 2002 and zero for all other time points, while step intervention is to assign the value zero before September, 2002 and assign a value of one for September, 2002–December, 2004.

We also analyzed the antibiotics data using GARCH models that take care of the heteroskedasticity of the data. When applied to total payments made for Erythromycin, the model built was fit much better compared to the ARIMA(1,2,0) model that was fit earlier. We used inflation rate as a dynamic regressor, which improved the model fit of the private insurance payment made for Cipro. Finally, we analyzed the data using text mining and clustering the ICD 9 codes that defined the patient condition, and made comparisons between the clusters formed on the severity of the disease. We used kernel density estimation to plot the six clusters on one plot for the antibiotic, Cipro, so that meaningful comparisons could be made.

CHAPTER 8

CONCLUSION

The aim of this dissertation was to develop time series models to investigate prescribing practices and patient usage of antibiotics with respect to the severity of the patient condition. We have analyzed factors that contribute to the cost of antibiotics such as total payments and private insurance payments. The amount of money spent varies between the severity of the patient condition; that means that patients who are more severely ill spend more money for antibiotics compared to patients who are less ill. Our goal was to develop time series models so that we can compare between several antibiotics. To reach this result, time series models and data mining tools such as text mining were used to investigate the prescription of antibiotics.

The main analysis idea was to show that the increase in the cost of antibiotic is affected by the increase in private insurance payment, and Medicare and Medicaid payments made. There was also a tremendous change in the number of prescriptions in one antibiotic compared to others.

We also investigated a sudden change, which we call it an outlier or event in time series theory. We found out that an outlier or an event does change the predicted payments made for the antibiotics. We also studied the effect of the inflation rate in cost of antibiotics, and found out the price prediction goes down. Since most of the antibiotics we studied are already available in generic form, or did not change to generic form during the time period under study, we could not take the switch in type into consideration. However, such a switch would involve a step function.

We also found out that the trends in the prescription of antibiotics were increasing over the years 1996-2004. There is a difference in the number of prescriptions of one antibiotic over another, with Amoxicillin mostly prescribed and Vancomycin least prescribed. We also investigated how much patients are spending on the sum and average for antibiotics and found out the spending was increasing over the years, but increasing less than the inflation rate. We examined the average Medicare and Medicaid payments for the antibiotic, Cephalexin, and we found out that the predicted Medicaid payment is three times as large as the Medicare payment. In terms of expenditure, the government's expense on Medicaid was increasing from the model forecast.

Text mining was used to reduce a large number of patient condition codes into an index of 6 levels, and to use those levels to examine the relationship between total payment and private insurance payments for different cluster levels.

The results of this study can be used by health care institutions and pharmaceutical companies to predict and forecast the distribution, cost, number of prescriptions, quantity of prescription dose, private insurance payment, Medicare payment and Medicaid payment. The trend of prescription can be used to study what the effect really is on the decrease or increase in the prescriptions of antibiotics.

REFERENCES

1. Mol PGM, Wieringa JE, NannanPanday PV, et al (2005). Improving compliance with hospital antibiotic guidelines: a time-series intervention analysis. *Journal of Antimicrobial Chemotherapy.* **55**:550-557.
2. Cerrito P, Badia A, Cerrito J (2005). Data mining medication prescriptions for a representative national sample. Paper presented at: PharmaSug 2005, 2005; Phoenix, AZ.
3. Cerrito P. Comparing the SAS Forecasting system with PROC HPF and Enterprise Miner. Paper presented at: SUGI 30, 2005; Philadelphia, PA.
4. Web access: http://www/meps.ahrg.gov/mepsweb/, May-2007.
5. Anonymous, (2006a). 2006 ICD9-CM and Medical Terminology Dictionary. ICD9.chrisendres.com.Retrieved, 2006, from the World wide web:http://icd9cm.chrisendres.com/index.php? action=contents. May-2007
6. Web access: http://www.medterms.com/script/main/art.asp?articlekey=33073 May-2007
7. Web access: http://www.meps.ahrg.gov/mepsweb/data_files/publications/st158.pdf May-2007.
8. Web access: http://www.meps.ahrg.gov/mepsweb/data_files/publications/st144.pdf May-2007.

9. Web access: http://www.emedicinehealth.com/antibiotics/article_em.htm, May-2007.
10. Chatfield, C. and Yar, M. (1988), "Holt-Winters Forecasting: Some Practical Issues," *The Statistician*, **37**, 129-140.
11. Aldrin, M. and Damsleth, E. (1989), "Forecasting Non-seasonal Time Series with Missing Observations," *Journal of Forecasting*, **8**, 97-116.
12. Al Migliaro and C.L.Jain (1984). Understanding business forecasting. Graceway publishing company, pp 69.
13. Ngai Hang Chan (2002). Time Series Applications to Finance, John Wiley & Sons, Inc.
14. Box, G.E.P., and Jenkins, G.M. (1976), *Time Series Analysis: Forecasting and Control, Revised Edition*, San Francisco: Holden-Day.
15. Chatfield, C. (1980), *The Analysis of Time Series: An Introduction, 2nd Ed,* London: Chapman and Hall.
16. Akaike, Hirotsugu (1974). "A new look at the statistical model identification". *IEEE Transactions on Automatic Control* **19** (6): 716–723.
17. Schwarz, G. (1978). Estimating the dimension of a model. Ann. Statist, **6**, 461-464.
18. Dickey and Fuller, (1979). Distribution of the estimator for autoregressive time series with a unit root, *J. Amer. Statist. Assoc.* **74** (1979), pp. 427–431.
19. Dickey, D.A., Hasza, D.P., Fuller, W.A., (1984). Testing for unit roots in seasonal time series. J. Amer. Statist. Assoc. **79**, 355-367.

20. James D. Hamilton, (1994). Time series analysis. Princeton university press.
21. Web access: http://www.ischool.berkley.edu/~hearst/text-mining.html, May-2007.
22. Online Webster-dictionary:http://www.webster-dictionary.org/, May-2007.
23. H. Karanikas, and B.Theodoulidis (2002), “Knowledge discovery in text and text mining software”, *Technical Report,* UMIST Department of Computation.
24. Miller, T.W (2005). Data and text mining, a business applications approach. Pearson Prentice Hall, New Jersey.
25. Kohnen, T.(2001), self organization maps, Springer.
26. Salton, G. and Buckley, C. (1998)“Term-weighting approaches in automatic text retrieval”, Information Processing and Management, **24** (5), 513-523.
27. Cerrito, Patricia (2006). Introduction to Data Mining with Enterprise Miner, Cary, NC; SAS Press.
28. Web access; http://jtac.uchicago.edu/conferences/05/resources/V&V_macal_pres.pdf, May- 2007.
29. Bollerslev, T. (1986), Generalized autoregressive conditional heteroskedasticity, Journal of Econometrics, **31**, 307-327.
30. Engle,Robert (2001). GARCH 101: The Use of ARCH/GARCH Models in Applied Econometrics, Journal of Economic Perspectives **15**: 157-168.

31. Engle, Robert (1982). Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation, Econometrica **50**:987-1007.
32. Tesfamicael, Mussie (2007). Dataminning to investigate the prescribing of medications longitudinally, SUGI conference, Orlando, FL.
33. SAS help and documentation, SAS Institute Inc., Cary, NC.
34. Web access: http://inflationdata.com/inflation/inflation_rate/InflationCalculator.asp , June-2007.
35. Tesfamicael, Mussie (2005). Calculation of health disparity indices using data mining and the SAS Bridge to ESRI, MWSUG conference, Cincinnati, OH.

CURRICULUM VITAE

NAME: Mussie Angesom Tesfamicael

ADDRESS: 4109 Ballard Avenue # 2
Cincinnati, OH 45209

DOB: Asmara, Eritrea-Jan 14, 1975

EDUCATION
&TRAINING: B.S., Mathematics
Asmara University
1993-1998
M.S., Mathematics
Southern Illinois University at Carbondale
2002-2003
MSPH, Biostatistics
University of Louisville
2003-2007

Comparison of Free and Commercial Sample Size Software Packages

PROFESSIONAL SOCIETES: American Statistical Society
American Mathematical Society

NATIONAL MEETING PRESENTATIONS:

- South East SAS Users Group, Oct 2004, Nashville, TN.
- Midwest SAS Users Group, Oct 2005, Cincinnati, OH.
- South East SAS Users Group, Oct 2006, Atlanta, GA.
- M2006 Datamining Conference, Oct 2006 Las Vegas, NV

INVITED PRESENTATIONS: SAS Global Forum, April 16-19, Orlando, FL.

PUBLICATIONS:

- Structural equation modeling assessing micro array data, Nashville, TN 2004, SESUG.
- Calculation of health disparity Indices Using Data Mining and the SAS Bridge to ESRI, Cincinnati, OH 2005.
- Gene Expression Profiling of DNA Microarray Data using Association rule and structural equation modeling, Atlanta, GA, 2006
- Datamining to investigate the prescribing of medications longitudinally, Las Vegas, NV, 2006

Currently working at Kendle International Inc. as a Statistical Programmer.