Projecting Future Medicaid Costs

New England Actuarial Seminars
VEE: Time Series
Winter 2013
Student Project
Benjamin A. Roberts

Introduction

Medicaid is a complicated government welfare program. At our company, we spend hundreds of hours per quarter to prepare a one to two-year Medicaid expense forecast for just one of the fifty states. This forecast is very detailed and takes into account the known political changes coming to the program as well as trends within each of several Medicaid subprograms.

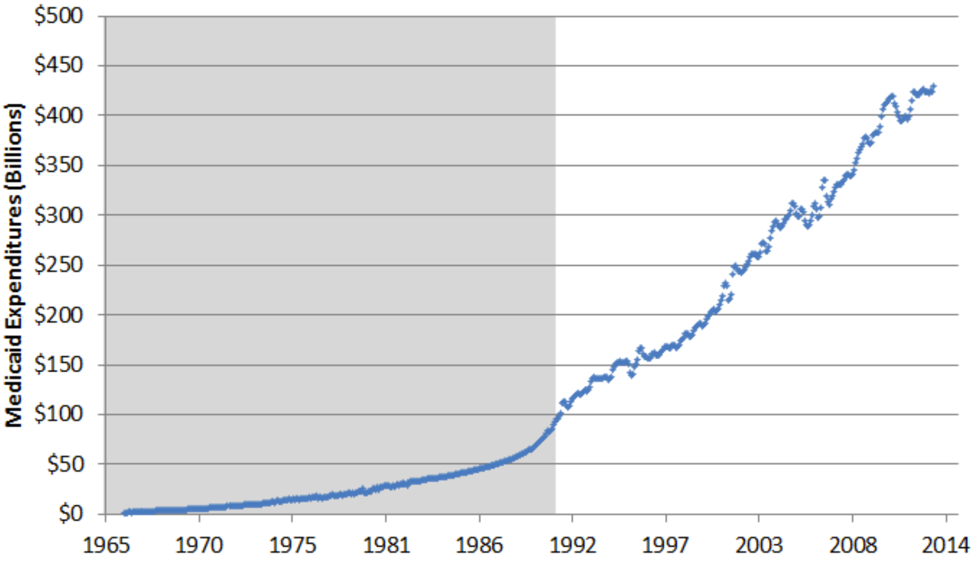
With this project, we want to take a broader look at national Medicaid costs. Instead of aggregating the forecasted effect of the many small parts of Medicaid, this project looks only at historical total Medicaid cost. It intends to identify a model that fits the historical trends in Medicaid costs enough to make a reasonable several-month forecast while avoiding the time-consuming headache of a thorough forecast. We also hope to establish bounds for what we expect the program to cost in the future.

To build the forecasting model we use Medicaid cost data released by the Federal Reserve Bank of St. Louis as well as population data released by the Census Bureau. To start, we take steps to achieve stationarity while preserving the interpretability of the time series as much as possible. Finally, after removing the seasonality of the first differences, we select an ARIMA(5,1,4) model. This is the model that we use for making our projections.

Data

At the time that we started the project, the most up-to-date Medicaid cost data on dataMarket.com was through July, 2013. This data shows annualized spending rates for every month since the inception of Medicaid in January, 1966. (By annualized spending rates, we mean that the values shown are monthly values multiplied by 12 so that what is plotted has the magnitude of an annual cost). Plotting the data immediately revealed that there were at least two regions to the data (Figure 1).

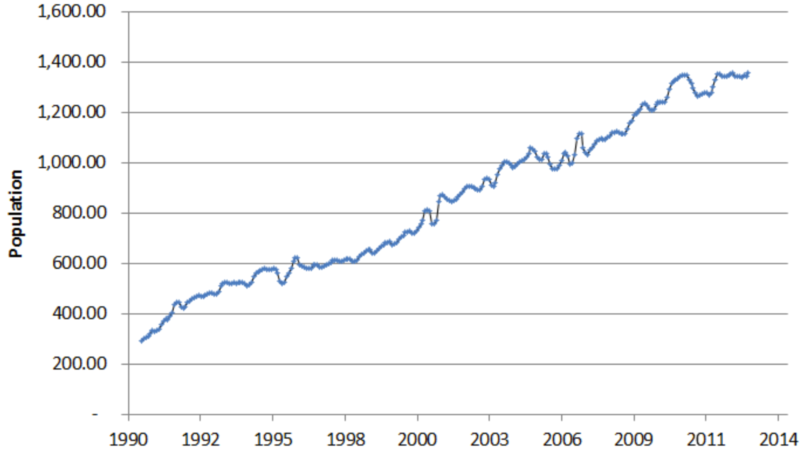
Figure 1: Total Medicaid Expenditures



Since we were interested in forecasting into the near future, we chose the more recent region of the data to analyze. The first record in our selected time series is for June, 1990.

We then began conditioning the data for forecasting. It obviously lacked stationarity since it trends dramatically upward. Before performing any analytics on the data, we wanted to remove this trend. We decided that one way to remove some trend might be to look at the expenditures on a per American basis. To do this, we took population data from Census.gov and divided each element of our cost time series by each corresponding element of our population time series (this calculation is performed on the "data" tab of the supporting Excel workbook). This ratio of Medicaid cost per American became our new "raw" data.

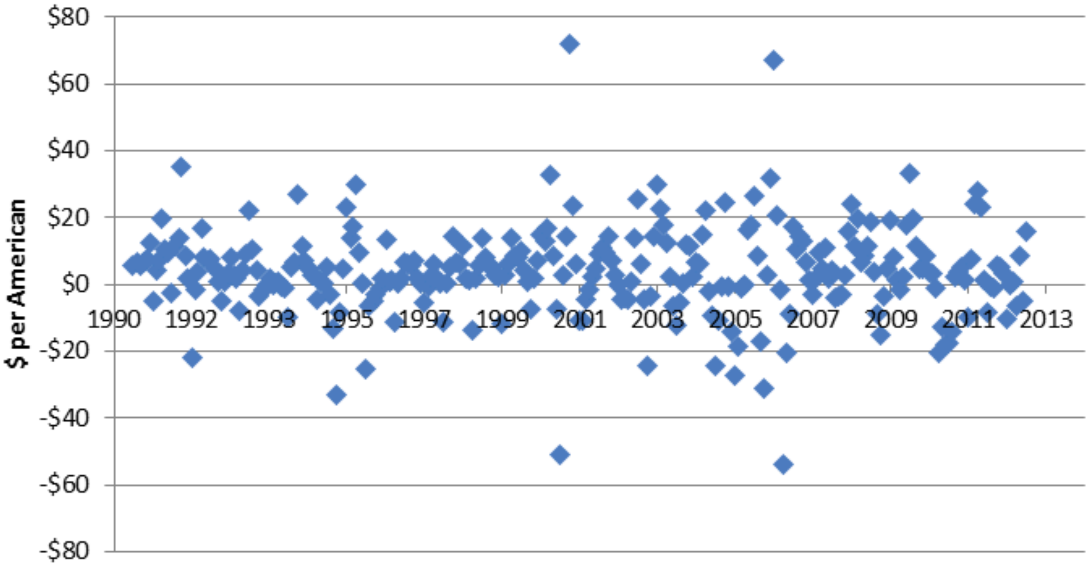
Figure 2: Medicaid Cost Per American



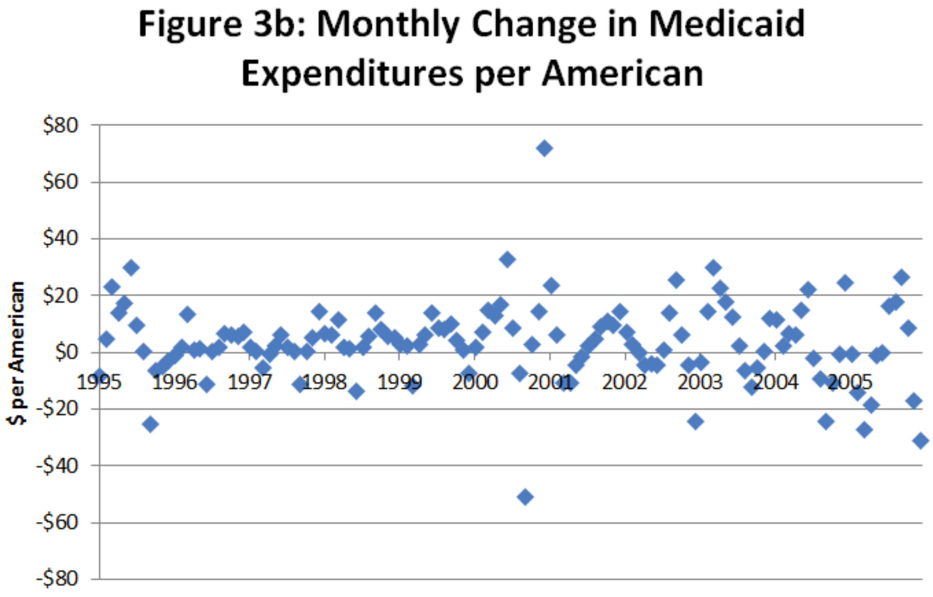
Data Conditioning

At this point, our data was still not stationary. Thus, we took first differences of the data (Figure 3a). After this, the data looked stationary, so we performed a stationarity test and confirmed this (the Figure 3 tab in the Excel workbook contains the results of the test at this stage).

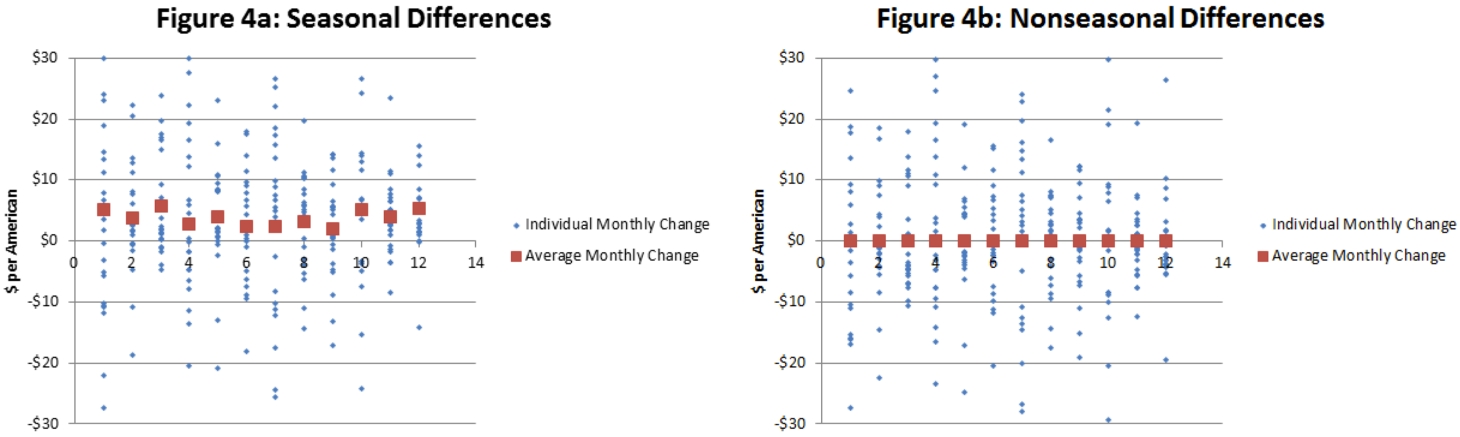
Figure 3a: First Differences of Medicaid Expenditures per American



As a final step before fitting an ARMA model to these first differences, we decided to look at seasonality. Though the data sometimes seemed to exhibit seasonality, it was unsteady at best (Figure 3b shows a subset of Figure 3a so that you can see potential seasonality).



When we took the average first difference for each of the 12 months, we discovered that there were slight seasonal trends. January, March, October, and December (months 1,3,10, and 12 shown in figure 4a) were the most expensive months while June, July, August, and September (months 6,7,8, and 9) were the least expensive. To deseasonalize the data, we subtracted a seasonality term from each data element so that the average first difference for each month was 0 (figure 4b).



Topic

All that has been described so far is just the process that we did in order to get a stationary, fittable data set. Before we cover some of the analytics we use to justify our modeling, perhaps it is worthwhile to take stock of what we have and how it will help us reach our goal. In a nutshell, our time series is the deseasonalized, monthly change in the annual spending rate on Medicaid per American. To achieve our goal of predicting the monthly cost of Medicaid per American, we will have to take a few steps to reconstruct our target values from the model projections.

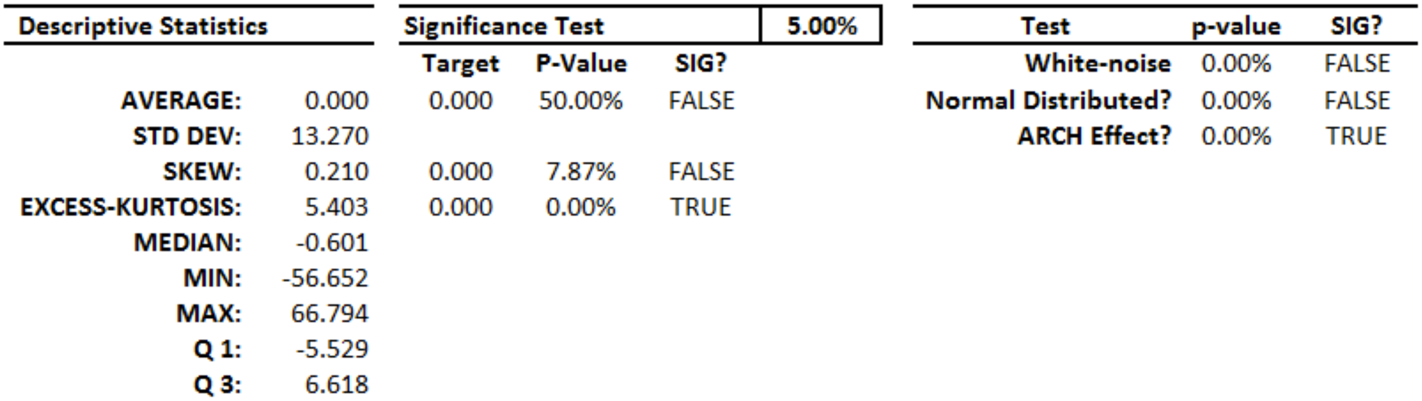
After using the model to make a projection, we will have to reseasonalize the data, adding back in the monthly averages that we subtracted in the data conditioning stage. At this point, the values we have are the projected month-to-month changes in Medicaid cost. To project the actual Medicaid cost, we will take the cost of the program as of the last month of our data and then subsequently add projected changes to find each projected cost.

Statistical Techniques for Model Fitting

Descriptive Statistics

After resolving seasonality, we looked at some summary statistics. While these statistics don’t help us with model building very much, they did help us to get a better idea of some of the time series’ key properties (This analysis takes place on the Figure 4 tab of the Excel workbook). First off, we can also see that the average value of the time series is 0; this is not surprising since we set this to be true in our deseasonalization process.

These descriptive statistics also include a series of tests. The section labelled significance test shows whether or not we believe our target statistics to hold for the data. For the “Average” test, the test is whether or not the average value of the data is 0. This is a two sided test, so we can think about doubling the P-Value and seeing how it compares to the significance level. Since double the P-value is 100% and the significance level is 5%, we have no reason to overrule our null hypothesis that the average of the time series was 0.



Descriptive Statistics		**Significance Test**		**5.00%**
		Target	**P-Value**	**SIG?**
AVERAGE:	0.000	0.000	50.00%	FALSE
STD DEV:	13.270			
SKEW:	0.210	0.000	7.87%	FALSE
EXCESS-KURTOSIS:	5.403	0.000	0.00%	TRUE
MEDIAN:	-0.601			
MIN:	-56.652			
MAX:	66.794			
Q 1:	-5.529			
Q 3:	6.618			

Test	**p-value**	**SIG?**
White-noise	0.00%	FALSE
Normal Distributed?	0.00%	FALSE
ARCH Effect?	0.00%	TRUE

The skew statistic is interesting in that the double of it's p-value (15.74%) is closer to the significance level of 5%. This means that the data does seem to have some skew, but no so much as to overrule the null hypothesis that the data are not skew. More evidence of slight skew is in the values shown for the minimum, Q1, median, Q3, and max. The max is greater than the negative of the min and Q3 is greater than the negative of Q1 while the median is slightly less than 0. This implies that the data has a slightly longer tail on the positive end and has more elements on the negative end. Still, the mean is 0. This clearly indicates some skew, but not enough to go against the null hypothesis that the data is not skew.

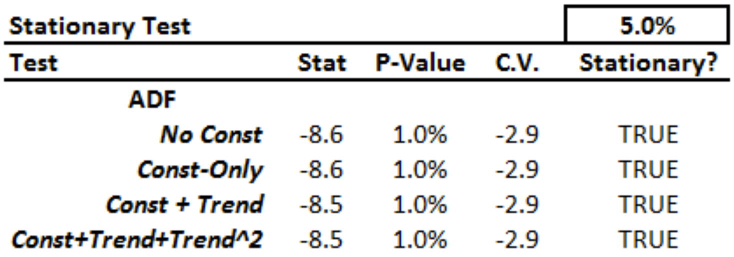
The excess-kurtosis test indicates that the kurtosis of the data is significantly more than that of a normal distribution.

The three tests to the right of the display above provide the most insight into what kind of model is needed for this data. The white-noise test is performed using Ljung-Box test method. This method will be explained later when we evaluate the model residuals. The null hypothesis of this test is that the data is white noise. Since we calculate a p-value of 0%, we conclude that the data is not white noise. This is good for modeling since it tells us that there are correlations in the data which will allow us to make better predictions.

The normality test assumes that the data follows a normal distribution. We would expect this to be false, given that the data has excess kurtosis. The final test shown on this exhibit is the ARCH test. It is testing for Auto Regressive Conditional Heteroskedasticity, meaning that it tests whether the data has volatility that is auto correlated with the data. The null hypothesis of this test is that there is no ARCH effect. Since the result of testing our data is "TRUE," it is probably best to fit the data to a GARCH model instead of ARIMA. We will overlook this fact, and proceed with fitting an ARIMA model, since that is more appropriate for the time series course that this project is intended to satisfy.

Stationarity

Before assuming that we can model the time series with an ARIMA model, we needed to demonstrate that the time series was stationary. We did this with the Augmented Dickey-Fuller test that is built into the Excel add-in that we used. This test uses several different test statistics and tests against the null hypothesis that the data is not stationary. Since we computed a P-value of 1% for each of the test variants, we conclude that the time series is stationary. (This summary is found on the Figure 4 tab of the Excel workbook)

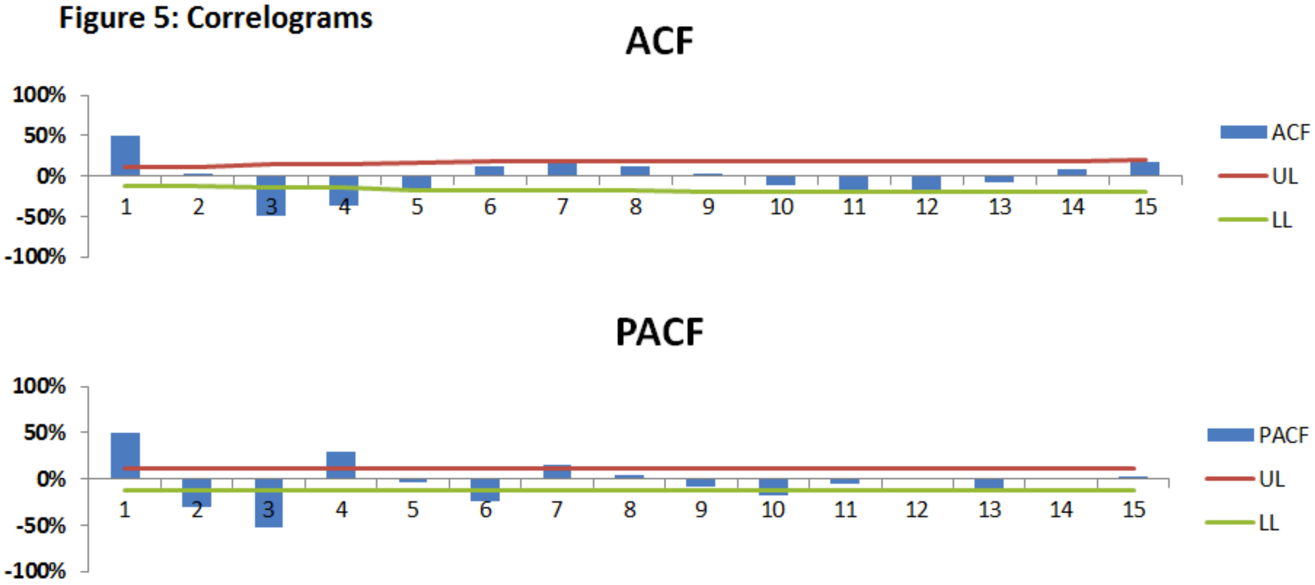


Stationary Test				**5.0%**
Test	**Stat**	**P-Value**	**C.V.**	**Stationary?**
ADF				
No Const	-8.6	1.0%	-2.9	TRUE
Const-Only	-8.6	1.0%	-2.9	TRUE
Const + Trend	-8.5	1.0%	-2.9	TRUE
Const+Trend+Trend^2	-8.5	1.0%	-2.9	TRUE

Correlogram

At this point, the groundwork was laid for building an ARIMA model. We used a correlogram to determine what sizes of models we should test.

Figure 5: Correlograms



The graph of partial autocorrelation function shows that there could be significant partial autocorrelation with the 10th lag, but that the number of autoregressive terms would likely be between 3 and 6 since there are relatively large autocorrelations with the third, fourth, and sixth lags.

On the correlogram depicting the autocorrelation function, we see very significant correlation with the third and fourth lags. We conclude that we will likely have 3 or 4 moving average terms. At the very least, we can see that we should have some of both kinds of terms in our ARIMA model. Both the partial and standard autocorrelation functions show very large correlations with the first lag.

Note: In the Excel workbook we build correlograms that looked at far more than 15 lags. The PACF graph had especially large correlations with large lags. We did not pursue these since it seemed unreasonable that the cost of Medicaid 10 years ago would be more correlated with today's cost than last month's cost is correlated with today's cost. We assumed that these large correlations were statistical anomalies driven by the fact that our data was not long enough to justify looking at such large lags.

ARIMA Model Fitting

The process of model-fitting is the crux of the analysis that we performed. Based on the conclusions we drew from our correlograms, we began testing various ARIMA models. With each model, we tweaked model parameters while observing the Log-Likelihood

Function (LLF) of the model, given the time series. This laborious task was actually performed by an optimization macro built into the Excel add-in that we used. However, each of the different models had to be set up manually. With each new model, we could compare the optimized LLF with LLF's from prior models.

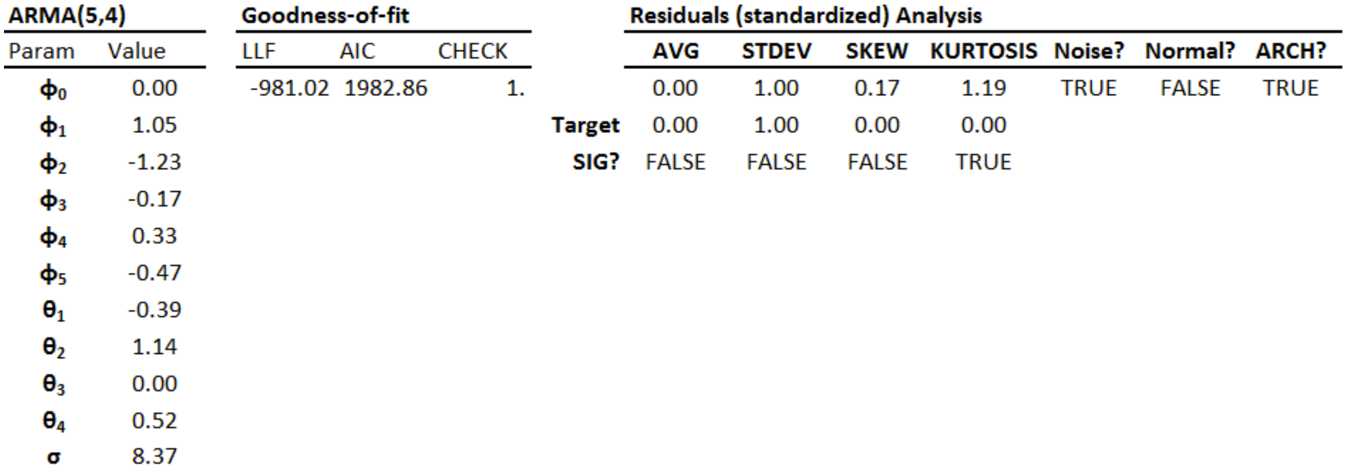
However, this comparison is biased since more regression terms will always allow you to increase the LLF. To decide which model to use, we instead optimized the Akaike-Information Criterion (AIC) which is calculated as:

$$AIC = -2 * LLF + \frac{2 * N * k}{N - k - 1}$$

where k is the number of model parameters and N is the number of data points. AIC is a systematic way of implementing the principle of parsimony--simpler is better--by taking on larger values for models with more parameters that don't improve the LLF. Note that the optimal AIC is the lowest.

After calibrating over 25 different models (this analysis is found in the Figure 6 tab of the Excel workbook), the optimal model was the ARIMA(5,1,4) with these parameters:

Figure 6: Selected Model



ARMA(5,4)

Param	Value
ϕ_0	0.00
ϕ_1	1.05
ϕ_2	-1.23
ϕ_3	-0.17
ϕ_4	0.33
ϕ_5	-0.47
θ_1	-0.39
θ_2	1.14
θ_3	0.00
θ_4	0.52
σ	8.37

Goodness-of-fit

LLF	AIC	CHECK
-981.02	1982.86	1.

Residuals (standardized) Analysis

	AVG	STDEV	SKEW	KURTOSIS	Noise?	Normal?	ARCH?
	0.00	1.00	0.17	1.19	TRUE	FALSE	TRUE
Target	0.00	1.00	0.00	0.00			
SIG?	FALSE	FALSE	FALSE	TRUE			

This ARMA model (which is an ARIMA model since it is modeling first differences) could be written formally as.

$$\widehat{W}_t = 1.06 * W_{t-1} - 1.24 * W_{t-2} - 0.16 * W_{t-3} + 0.31 * W_{t-4} - 0.47 * W_{t-5} + 0.39e_{t-1} - 1.15e_{t-1} + 0.01e_{t-1} - 0.53e_{t-1} + e_t$$

It is interesting to note that, based on the standard residual analysis shown in the exhibit above, the residuals has some of the same characteristics as the time series that the model was used to fit. The average is 0 and there is some skew, but not a statistically significant amount. Furthermore, the residuals still have significant excess-Kurtosis which appears to be related to the ARCH effect. The residuals do not follow a normal distribution, but unlike the time series data, the residuals do pass the white noise test. This is different and worth looking at more closely with a thorough Ljung-Box test.

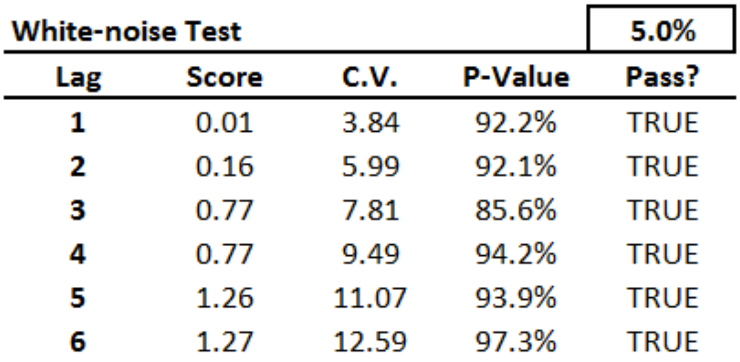
White Noise of Residuals

For our analysis, we chose to use the Ljung Box test which is what was recommended by the Excel add-in that we used for our analysis. Fortunately, the Ljung Box test is easily constructed. When testing m lags of T data points for a time series, the test statistic is:

$$Q^{*} = T(T+2)\sum_{j=1}^{m}\frac{\hat{\rho}_j^2}{T-1}$$

Where rho(j) is the sample autocorrelation at lag j. This sampling distribution of this test statistic chi-square with m degrees of freedom and the null hypothesis is that the time series is not serially correlated. The documentation the time series software that we used describes the white noise test as a statistical portmanteau ljung box test. This means that instead of just testing all the maximum number of lags that you are concerned with, you run several tests, starting with just the first lag, then the first two lags, and then the first three lags, until you reach the maximum number of lags that you want to test. In our analysis, we tested the first 6 lags since the test begins to loose its power when testing over ln(T) lags (according the the documentation for our Excel add-in). In our case, ln(T) is about 5.6.

Figure 7



White-noise Test				5.0%
Lag	**Score**	**C.V.**	**P-Value**	**Pass?**
1	0.01	3.84	92.2%	TRUE
2	0.16	5.99	92.1%	TRUE
3	0.77	7.81	85.6%	TRUE
4	0.77	9.49	94.2%	TRUE
5	1.26	11.07	93.9%	TRUE
6	1.27	12.59	97.3%	TRUE

None of the tests indicate a lack of white noise. Based on the p-values which are all in the 90%'s, we can see that we don't come close to the significance level of 5% that we require to overturn the null hypothesis that the time series is a white noise process.

Corrections and Adjustments

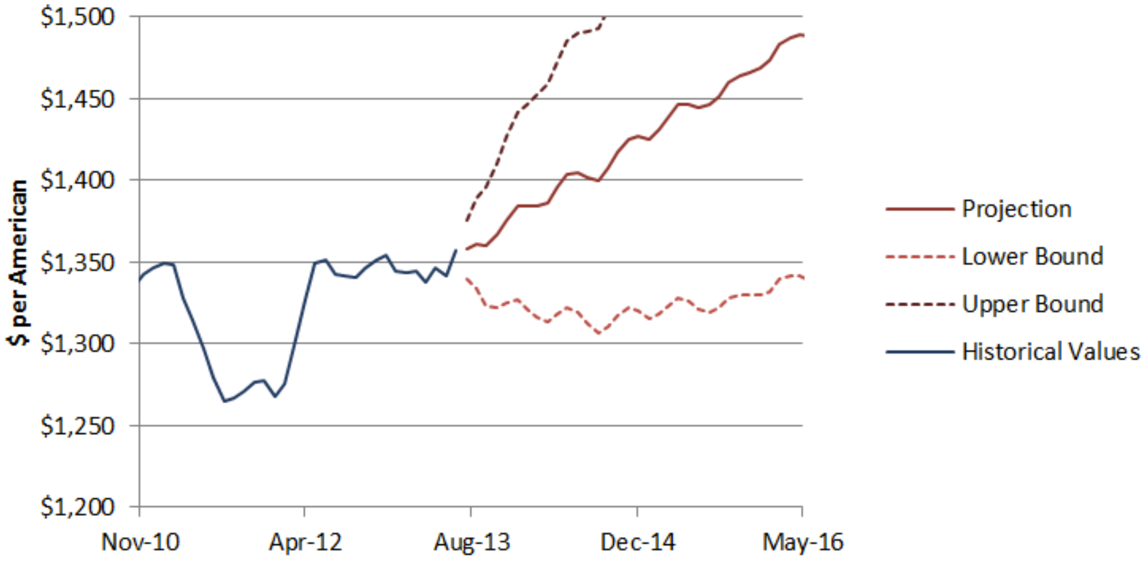
The most significant corrections and adjustments were already discussed earlier in the project. The first adjustment that we made was to use only the second half of the Medicaid expenditures data. Another adjustment that we made was to remove some seasonality from the data.

Projections

The final stage of our project is to make a prediction about the future. The most difficult aspect of making a projection with this model is that our model did not succeed in producing normally-distributed residuals. This made it so that our 95% confidence interval for the projection could not be calculated using derivatives of the normal distribution such as the student-T distribution. We compensated for this in the Excel workbook (see the Figure 8 tab).

Another complicating factor was that the software that we used output standard deviations for each element of the projection, but these elements were first differences, not the actual values. To find the standard deviation of the projected medicaid cost, we took the square root of the sum of the variances of all the first difference terms leading up the the projected element. This summing effect made it so that we could assume the projection was normally distributed for later terms in the projection (by the law of large numbers). For the early terms in the projection, we used a blending of a normal distribution with the empirical distribution of our residuals. By the 50th projected term, we assumed that the projected terms are fully normally distributed.

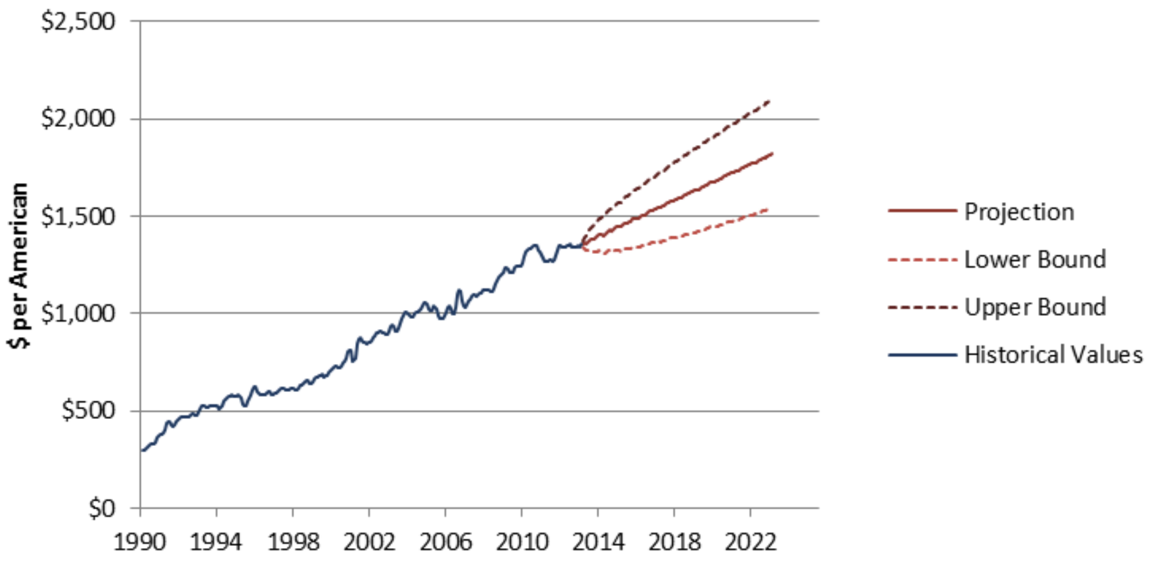
Figure 8a: Short-Term Medicaid Cost Projection (with 95% confidence interval)



Despite all of our efforts to take a robust approach to fitting the data, the standard deviations of the projected elements were substantial. This made it so that in the short term, there is a wide range of possible outcomes. When you see the volatility of the historical data, however, this is not very surprising.

The long-term projection yields a similarly broad range of possibilities for future Medicaid costs, but this is not surprising either.

Figure 8b: Long-Term Medicaid Cost Projection (with 95% confidence interval)



Conclusion

The results produced by our selected model are not at all surprising. Historically, the cost of Medicaid per American has tracked fairly linearly. Probably the biggest insight that this model has to offer is the projection of the range of possible outcomes calculated. For any government budgeting purposes, the range would likely be unacceptable. Still, to the average American, the results we found are straightforward and helpful in identifying future uncertainty.

References:

Data Sources

Medicaid Costs
Exported from datamarket.com on September 1, 2013.
http://datamarket.com/data/set/22kb/personal-current-transfer-receipts-government-social-benefits-to-persons#!display=line&ds=22kb!2ejq=1
Data provider: Federal Reserve Bank of St. Louis
Subset: Medicaid
Units: Billions of Dollars

Population Count
Exported from census.gov on September 7, 2013.
Data Provider: Census Bureau
Units: People

For data range 1990-2000
http://www.census.gov/popest/data/intercensal/national/files/US-EST90INT-07.csv

For data range 2000-2010
http://www.census.gov/popest/data/intercensal/national/files/US-EST00INT-TOT.csv

For data range 2010-2013
http://www.census.gov/popest/data/state/totals/2012/tables/NA-EST2012-01.csv

Excel Time Series Add-In

http://www.spiderfinancial.com
product: NumXL