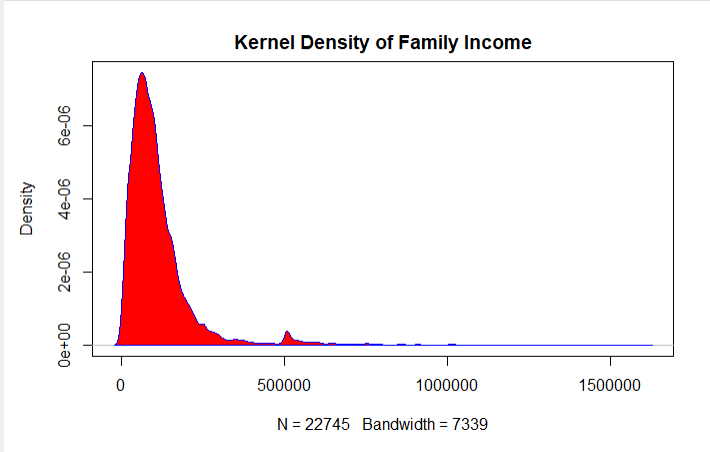
**Exercise 3: Predicting Household Income with Logistic Regression**

**Part 1: Background**

In this exercise, I am attempting to predict if a household has an income greater than $150,000 in New York state, given a subset of the data contained in the 2010 American Community survey for New York state. Since I’ll be testing (and creating) a binary variable, logistic regression is the best choice here. I’ll be determining which variables to include in my model and which to discard.

**Part 2: Method**

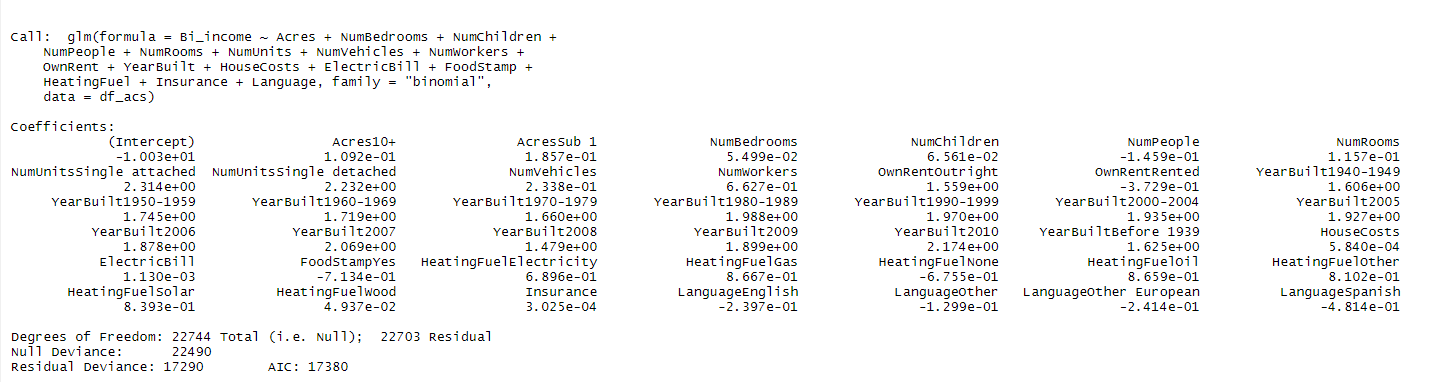
To start I first read in the csv data and created a dataframe, since dataframes are easier to work with. Then, I wanted to get an idea of how the Family Income data was distributed, so I created a Kernel Density plot:



As you can see, the vast majority of the data is concentrated around the $125,00-$250,000 range, with about $150,000 seeming to be the most common. Therefore, creating a model able to predict income values greater than $150,000 would be valuable. Next, in order to create a model predicting a yes or no answer on if a family’s income range was over $150,000, I needed to create a new binary variable column expressing this. This column would be the dependent variable for the exercise. I did this by creating a new column named “Bi\_income” at the end of my dataframe, using the “with()” function to set the cell value equal to 1 if a family’s income was over $150,000, and 0 if it wasn’t.

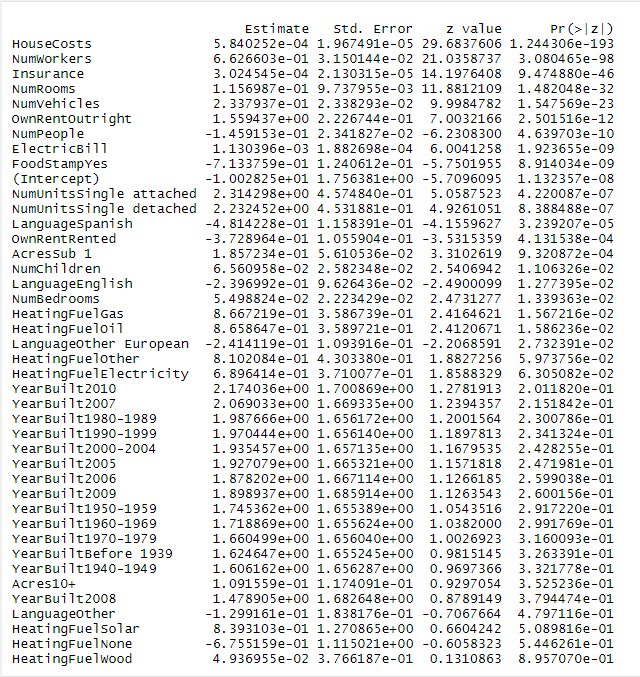
**Part 3: Running the Initial Logistic Regression Model**

Now I was able to perform the first logistic regression model test (named acs\_model), using the glm() function and setting the distribution type to “binomial.” Testing all variables at the start, here were my results:

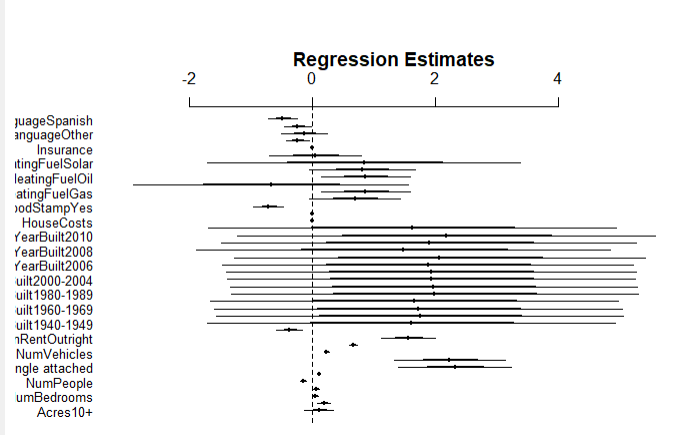


The important points here are Null Deviance, Residual Deviance, and AIC. Null deviance is the value of the null model, i.e. how well the model would predict the “Bi\_income” variable if the model only had the intercept and each data point was equal to the average of “Bi\_income.” The residual deviance is the value of the fitted model, where all the variables are included. Having the residual deviance lower than the null deviance means the model fits better with the variables, which is a good sign. Finally, AIC measures the quality of the model, and in short, estimates the relative amount of information lost by a given model. AIC is used to compare model quality between models. The lower the AIC, the better.

Next I tested how fitting the coefficients of the variables were, and if they should be included at all. The summary() function then gave me the coefficients and p-values for each variable. Also, since so many variables were used, I used some code to order the variables by p-value in ascending order. Here were my results:



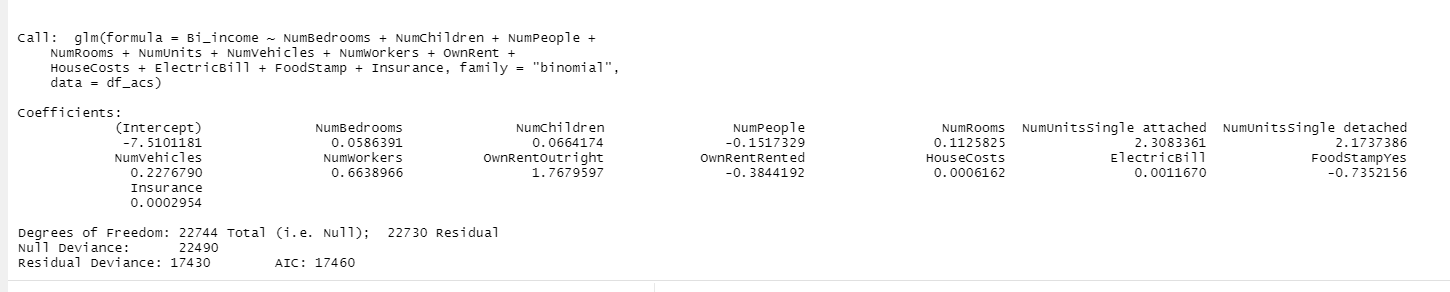
As you can see, all the p-values for each variable coefficient are listed. Also, in columns where the data was categorical, R created them as dummy variables. In deciding whether to keep the variables or not, I used a p-value (listed as Pr(>|z|) of 0.05. If the p-value was greater than 0.05, I removed the variable from the model. A problem though was that some dummy variables for a given category were below this threshold while some were above it. From my research of best practices, I determined that I should either remove the all the dummy variables of a specific column or none of them. Since there are so many variables already, I decided to remove them. These removed variables ended up being Heating Source, Year Built, Acres, and Language. These removals were also justified by the coefficient plot for this regression:



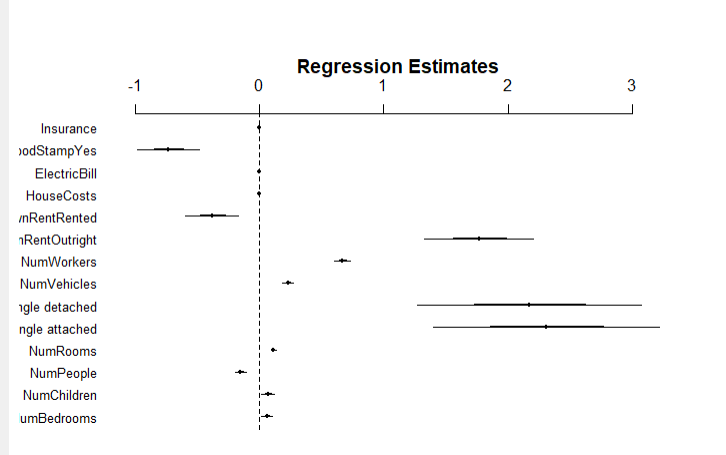
This shows the estimated variable coefficients along with their 95% confidence intervals. In general, variables whose confidence intervals intersect the reference line at 0 are not significant (since that means their coefficients could be 0, which mean they wouldn’t be included in the model.) All the variables we removed had confidence intervals spanning 0.

**Part 4: Refining Logistic Regression Model**

Once the insignificant variables were removed, the logistic regression model was reran (named acs\_model2), and these were my results:



Surprisingly, the AIC value went up. This indicates that this new model actually predicts if the family income is greater than $150,000 worse than the model with all the variables. This could mean that some of the removed variables were more significant than I thought (some dummy variables did have p-values under 0.05 after all). However, I still believe that the new model is better. The original model had far too many variables, and I believe there was a significant risk of overfitting. Also, while some of the dummy variables of a category had p-values under 0.05, it’d be bad practice to only use a variable if its value was equal to a specific amount- it has to be all or nothing. I can also feel justified in my new model by looking at the new coefficient plot for the new model:



Two aspects of this plot show why the new model is better. First, none of the variables have coefficients with confidence intervals crossing the 0 value. Therefore, all the variables are statistically significant to our model. Additionally, the confidence intervals for each variable in this plot are roughly equal or narrower than what they were in the original coefficient plot. This means that the standard error is lower for each variable, which increases the reliability of our model. Taking everything into account, I am confident in concluding that given the two logistic regression models, that the second model, “acs\_model2” is best.

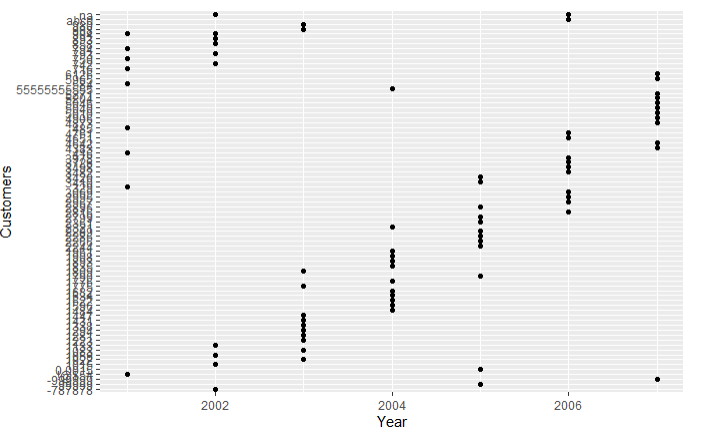
**Exercise 4: Predicting Patient Count for 12 Months for a Dental Clinic**

**Part 1: Background**

In this exercise the goal was to find the best way to predict the patient count for a year using data given in a csv file using forecasting techniques in R. The two forecasting techniques used were the Holt-Winters method and the ARIMA method. Analysis was then done on each one to determine which one was best. However, there was an issue with the data we were given: it wasn’t clean. This means that some of the cells had invalid entries that were either missing or didn’t make sense in context of the column (less than 0, extremely high outliers, gibberish, etc.). Therefore, we had to clean the data by turning these values to “NA” and then using an imputation method to fill in those values.

**Part 2: Cleaning and Imputation of Data**

After reading in the “BestSmileDental.csv” file, there was a number of issues with it, as visually seen by this scatterplot of the initial data:



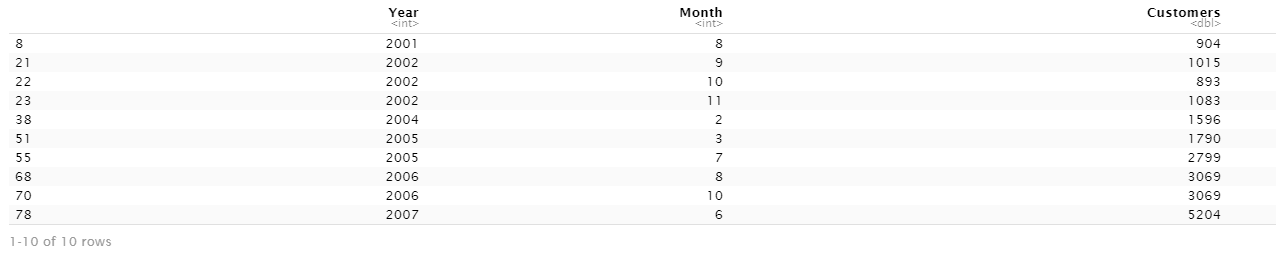
This scatterplot is almost incomprehensible. Looking closer at the data, I saw a number of issues in the Customers column: values less than 0, a value higher than the population of the world, string data instead of a number, and a fraction. A visual inspection of the Year and Month columns showed now invalid data. In order to use imputation methods, invalid data needed to be set to NA. So my first goal was to set these invalid data cells to NA. I figured the easiest way to do this was to just convert the data type of the Customers column to numeric. This changed all the string values to NA, also, allowed me to get summary statistics in order to check for any numbers that didn’t make sense. Doing so, I then saw a minimum value of less than 0. So I changed all values less than 0 to NA. Next, I set all values greater than 100,000 equal to NA. I choose 100,000 given the median value of the data set. I was confident any values above this were for sure incorrect. Finally, there was a decimal value, and since you can’t have a fraction of a customer, I set that to NA as well. Running the newly created dataframe with the NA values replacing invalid data, I saw that all the values were either valid or NA, which means it was ready for imputation.

Before imputing the NA values, I had to find where they were located, in order to ensure they were changed. Here I used the “which()” function to find the row numbers then called the rows:

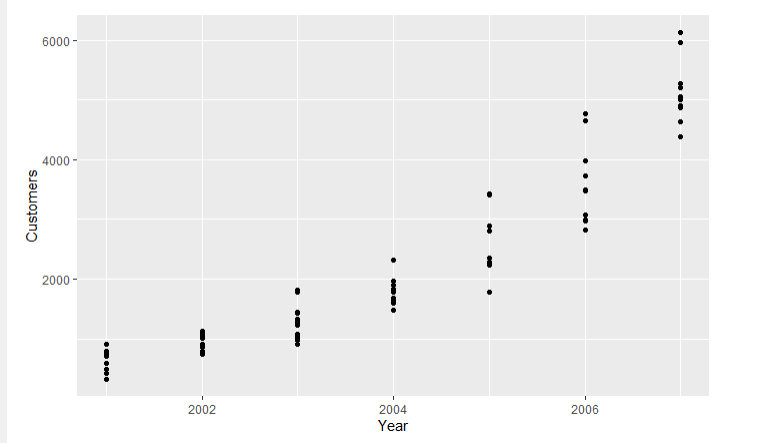
Table

Description automatically generated

Now I was ready to begin imputation. I chose to use the “mice” package because I saw it was a commonly used package and was well explained in the text book “R in Action.” Imputation in statistics is the process of removing nonexistent (or “NA”) values from a set of data and replacing them with values that fit the rest of the data. These values aren’t exact, and the “mice” package can run through imputations multiple times and choose different values. When I ran my test, I used the default value, which was the firs imputation set R used. After running the imputation, I called the rows that previously had NA values:



The imputation worked perfectly. Finally, I plotted a scatterplot with the new cleaned and imputed data, to ensure everything worked and there was no more cleaning to do:



The scatterplot looks much better now, and the data is ready to be forecasted.

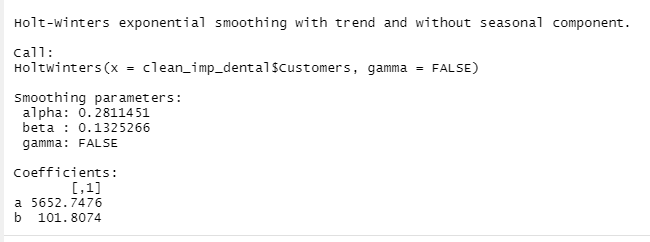
**Part 3: The Holt-Winters Model**

The Holt-Winters Model is one way to forecast data in R. Before running the test, I plotted the data as a time series to see what the parameters of the model were:

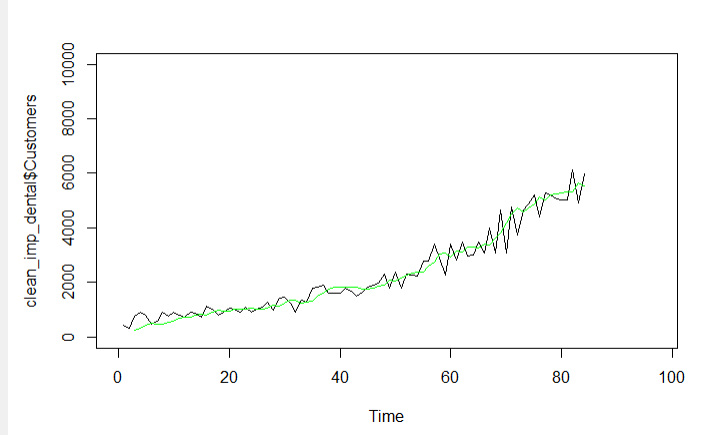
Chart, histogram, scatter chart

Description automatically generated

The arguments of the method I’m concerned about are alpha, beta, and gamma. I let R determine the alpha value in this set. For the beta value, since the graph had a trend of steadily increasing, I set the beta value to True. Finally, since the data didn’t appear to be seasonal at all, I set the gamma value to false. The results of the Holt-Winters test:



Then I plotted the best fit line of the data, which on upon visual inspection seems like a decent fit:

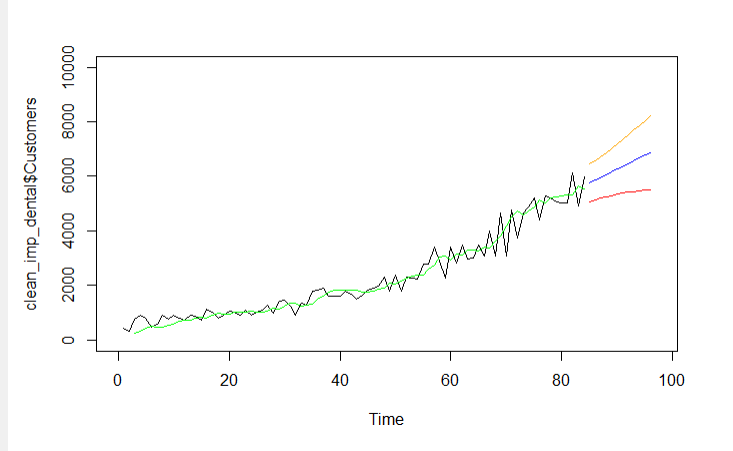


Once I did this, I predicted the next year (12 months) of customer values. This gave me not only the best fit values, but an upper estimate and a lower estimate:

A picture containing text, receipt

Description automatically generated

And I also plotted this data, with the green line being the best fit line of the data, the blue being the expected prediction values, the orange being the upper bounded prediction values, and red being the lower bounded prediction values.



In order to come up with the best model, I also used another method to forecast values, the ARIMA model.

**Part 4:** **The ARIMA Model**

The ARIMA Model, or autoregressive moving average, ARIMA uses a number of lagged observations of time series to forecast observations. In order to test if an ARIMA model is appropriate the ACF (Autocorrelation function) and PACF (Partial Autocorrelation function) are tested. Running the ACF value and PACF values, I got these graphs:

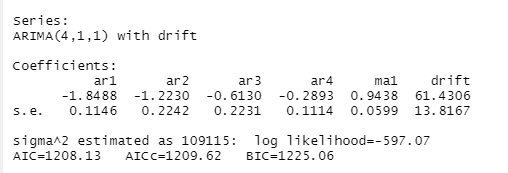
Chart

Description automatically generated

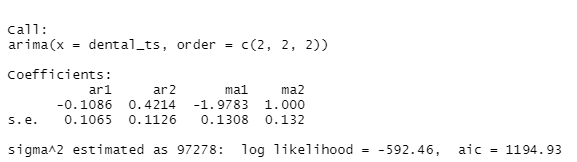
Chart, timeline

Description automatically generated

From the ACF graph, since the values start with a high spike and decrease, we can see there’s some autoregression in our data. From the PACF graph, since there are large spikes at the beginning, it indicates moving averages. Since we have autoregression and moving averages, ARIMA is an appropriate method for model forecasting. Now I could create my ARIMA model. At first, I used default values for p,d, and q, which ended up being 4, 1, and 1. Return the results gives me the following values:



However, playing around with the p,d,q values, I found setting the values to 2, 2, and 2 actually gave me a lower AIC value, so I went with that:



Next, to see how well this model works, I plot the ACF and PACF values of the residuals of this ARIMA generated model:

A picture containing chart

Description automatically generated

Here we can see that the first value equals 1, while every other value is within the blue lines. This is a good sign, as it’s indicative that the residuals are statistically insignificant, and means the model is more accurate.

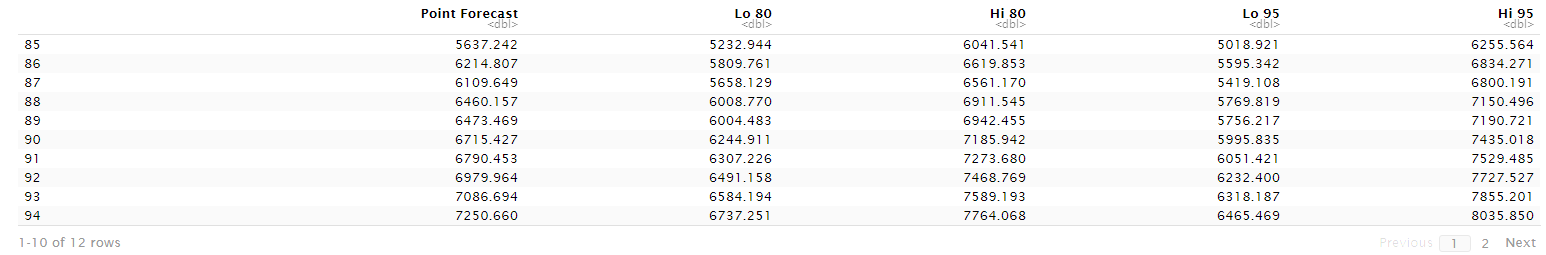
Chart, timeline

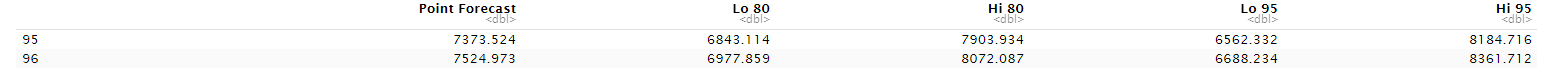
Description automatically generated

Like the ACF graph, having all values within the blue lines means the residuals are statistically insignificant and are a good sign of a valid model.

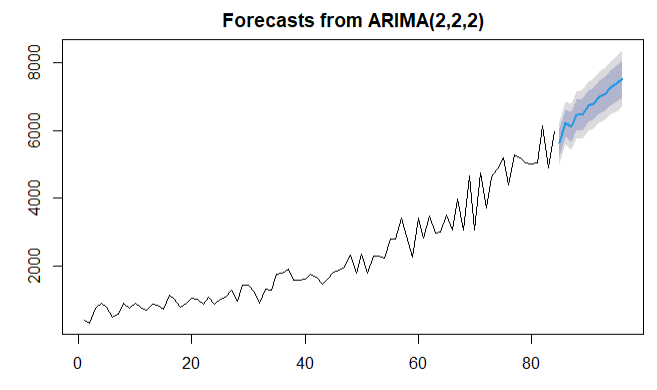
(Note: I also found the ACF and PACF residual plots of the manually entered p,d,q ARIMA model to be better than the auto generated ARIMA ones. For instance, the graph on the auto generated ARIMA model had residuals going above the blue lines.)

Finally, since I have a viable model of the given data, I’m ready to create a prediction of the Customer count for the next year. Below are the raw expected numbers of the prediction, along with the graph with the forecasted values:





Here we are given the expected forecast value, along with values for an 80% confidence interval and a 95% confidence interval.



A graph of the predicted values for the ARIMA method. The blue line is the expected predicted value, the dark grey area is the 80% confidence interval, and the light grey is the boundary for the 95% confidence interval.

**Part 5: Choosing the Best Model**

Upon visual inspection, I’d be inclined to say the ARIMA model is a better fit. The prediction line of the ARIMA model incorporates the jagged nature of the data, unlike the Holt-Winters model which is flat. Also, the confidence interval of the ARIMA model seems narrower than the confidence interval of the Holt-Winters model, so we have a better approximation of predicted values. However, to actually compare models in a statistical way, I used the “accuracy()” R function on the models to find error measurements, as seen here:

Text

Description automatically generated

The values I was most interested in and used to the determine which model was better were the MAPE values. The MAPE (Mean Absolute Percentage Error) value essentially measures the size of the error between predicted and forecasted data in percentage terms. The lower the MAPE value the better, and good models have MAPE values under about 20%. The MAPE is the primary indicator of forecast accuracy for many organizations.

Both the Holt-Winters and ARIMA models have values under 20%. However, the ARIMA model’s MAPE value is slightly lower. Therefore, my initial intuition was correct, and the ARIMA model should be the better forecaster of the number of Customers each month in the Best Smile Dental clinic for 2008.