**Weekly Sales Data for Walmart: Time Series Analysis**

Nathan Fritter

PSTAT 174

Perm: 6657761

TA: Zhou

Section: T 4pm

Date: Monday, June 6, 2016

**Introduction/Plotting the Original Time Series**

Companies from various industries and practices go through busy periods and slow periods throughout a year, and this also can result from seasonal increases and decreases in demand (for example demand might skyrocket during holiday times and decrease when holidays pass).

For this project I will be analyzing sales data for various Walmart stores and departments for multiple years (February 5, 2010 to October 26, 2012; weekly). The data set is very comprehensive and contains 45 stores with up to 98 departments each. There is other useful information as well for possible machine learning algorithms (employee data at each store/department, whether or not the week had a holiday, etc.), but those other factors will not be taken into account.

The aim of modeling here will be to predict weekly sales data for this specific store (I chose the first store, first department), taking seasonality and other factors into account. Here is the graph of the data as a time series object:

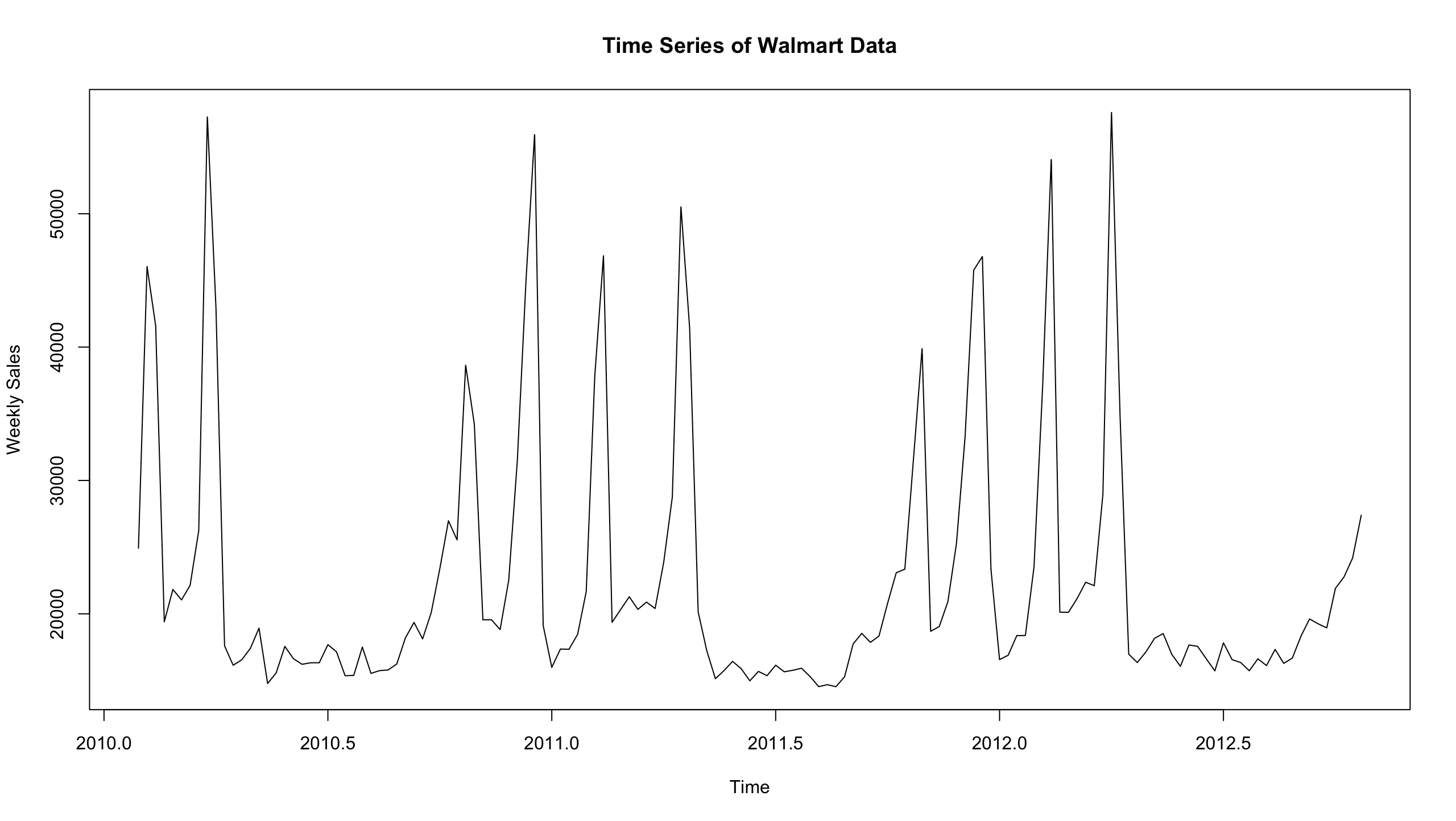


Figure 1: Time Series Graph of Walmart Sales Data

Looking at Figure 1, we can see that there is a ton of volatility going on with this data, and the peaks of the graph most likely coincide with the weeks containing holidays (and the lower parts after the holidays end). It also seems that the beginning and ends of the years are the most productive for the store/department (the “Time” axis is in weeks; weeks 0-10 and around weeks 45-50 have high sales with lower weekly sales in between). A lot of this can be attributed to seasonality.

A differencing transformation might be appropriate to improve stationarity; there does not appear to be much of a trend since the data moves up and down aggressively at certain times. And while the volatility of this data is high, the trend and seasonality aspects likely account for much of it. We will see how much these affect the data in the next step.

**Remove Seasonality and Trend**

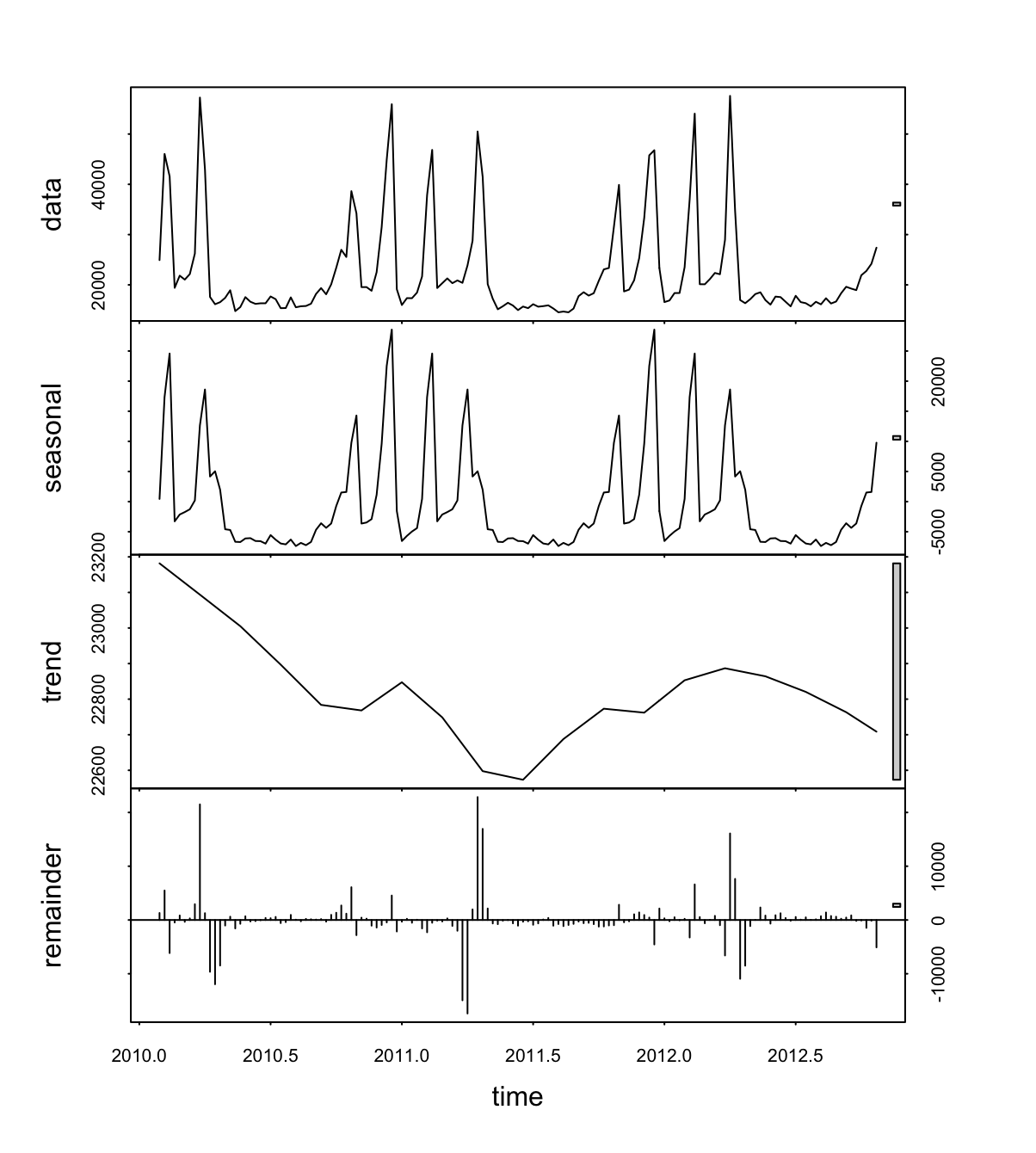


Figure 2: Decomposing the Original Data

Using the stl() function, I am able to decompose the data by splitting the original data into its seasonal, trend and random components (which is “remainder” above). The stationary residuals are at the bottom of the graph in Figure 2. Besides for some obvious seasonal components that continuously show up around the beginning of each year, we can see that the data is stationary (the remainder data hovers around zero and it is homoscedastic). A log transformation is not necessary because the data is not heteroscedastic; in fact, any such transformation would only marginally add to the strength of a model.

However, I will go ahead and difference the data as well as perform tests for the first difference of the data just to confirm this. Looking ahead to the graph on the next page (which is a time series plot of the differenced data):



Figure 3: Taking the First Difference of the Data

Looking at Figure 3 we see that differencing the data gives us nicer data seeming to be moving up and down around zero and does not have as much volatility as the original data.

Now I will turn the differenced data into a time series object and decompose the data to see if either the seasonality or trend has been lessened/removed:

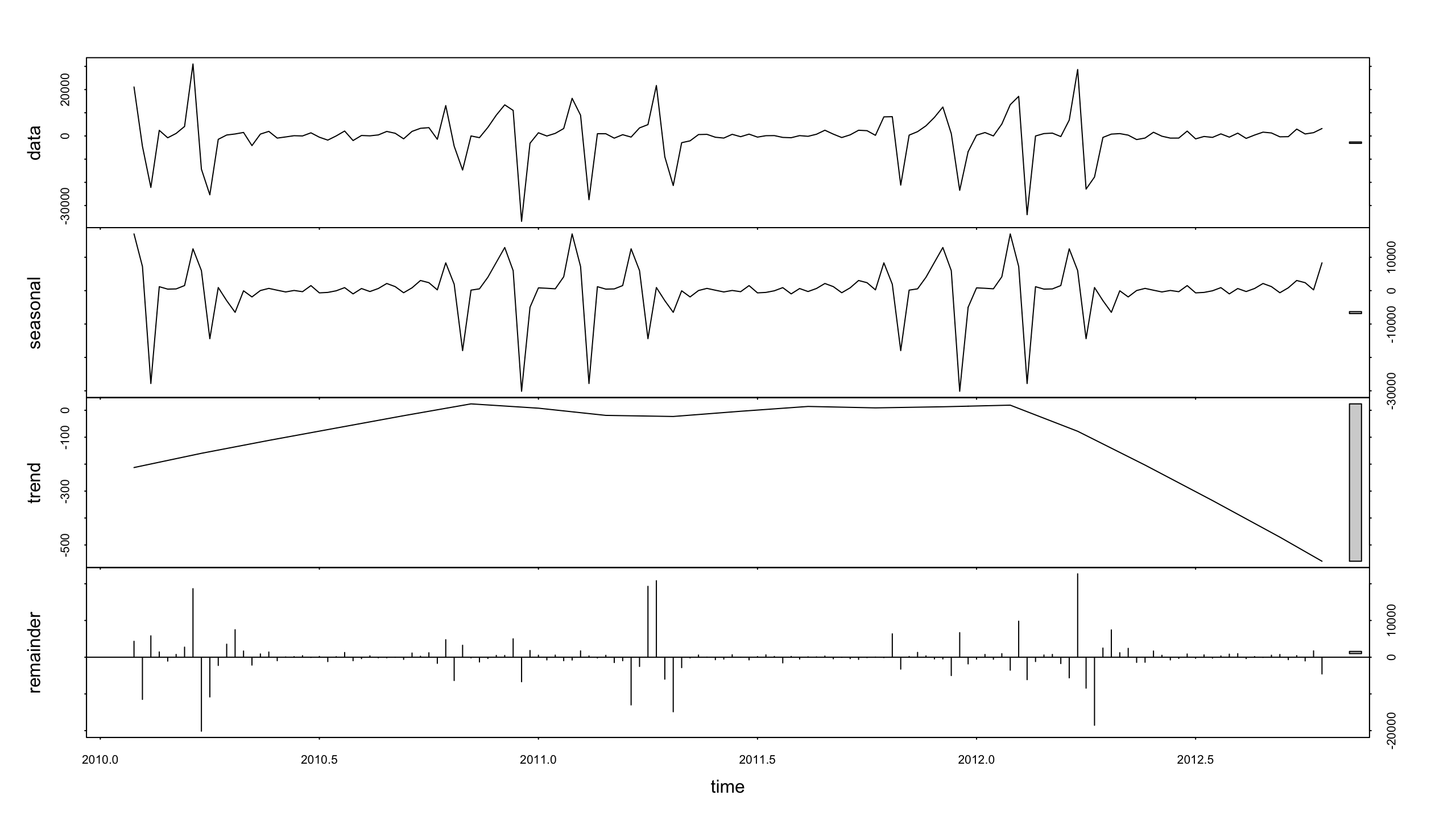


Figure 4: Decomposition of the Differenced Data

The important thing to take away from Figure 4 above is that the seasonal aspect of the differenced data is almost exactly the same as the actual differenced data. This means that there is such a strong seasonal component to my data that transformations really will not help that much. Instead, using the seasonal aspect of ARIMA models (P, D, Q) is the better way to find a good model.

**Plot and Analyze ACF and PACF**

Here is what the ACF and PACF look like for the original data:

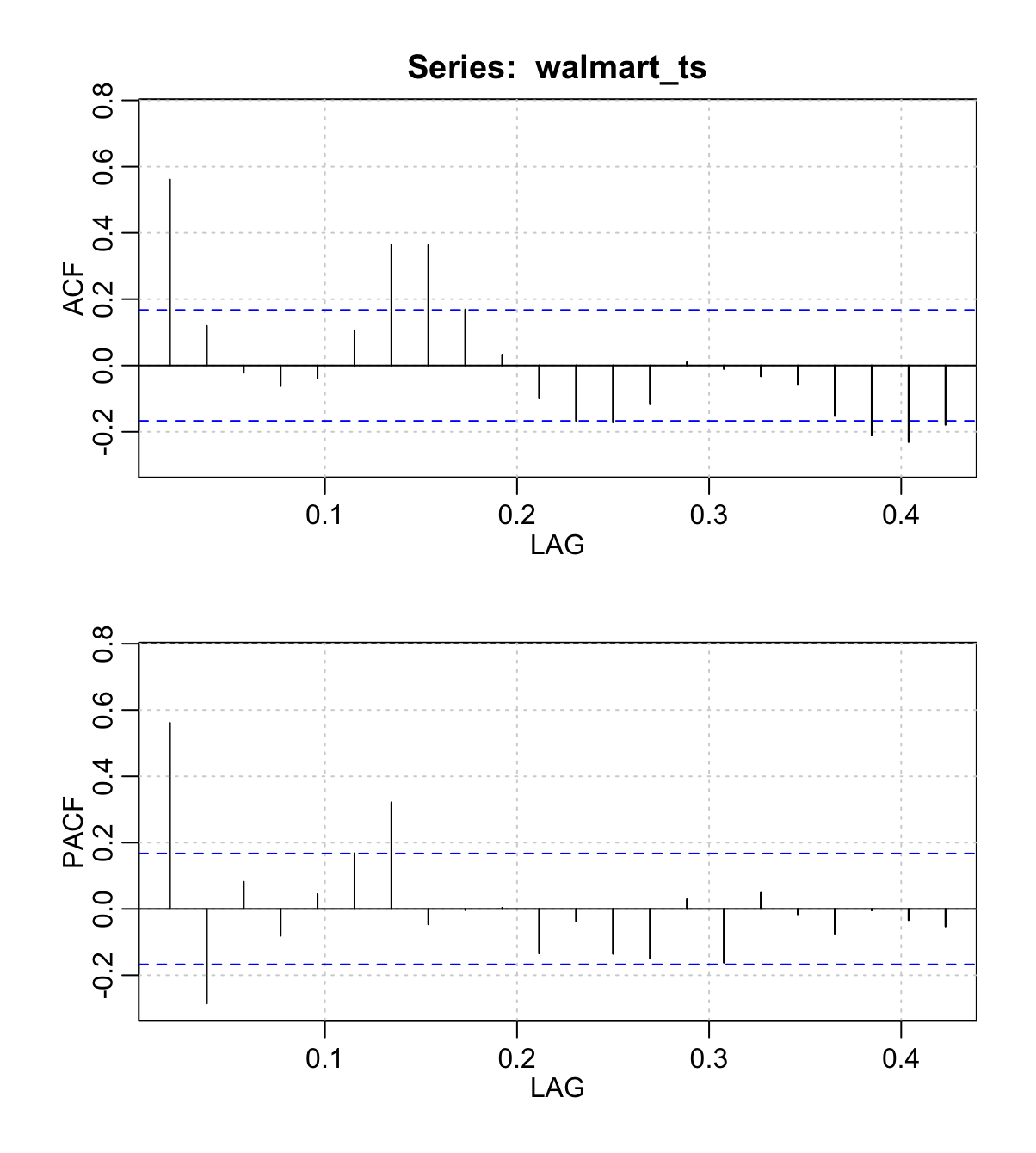


Figure 5: ACF & PACF Graphs

ACF stands for Autocovariance Function, and PACF stands for Partial Autocovariance Function. Unfortunately Figure 5 above does not give us much to work with. Since the ACF values do not cut off entirely at a certain lag, the time series cannot solely be a Moving Average (MA) model. Since the PACF values do not cut off entirely at a certain lag as well, the time series cannot solely be an Auto Regressive (AR) model either.

Thus, it must be an ARMA model; while neither ACF or PACF values exhibit strong geometric decay behavior one can see the values getting smaller overall even if they come back to increase.

With actual data, the ACF and PACF are considered “cut off” if at a certain point the lags fall back within the confidence interval lines. Thus we can see the ACF is “cut off” at lag 8 and the PACF is “cut off” at lag 7, with 3 lags in both the ACF and PACF outside the blue dotted lines.

This means there is an Auto Regressive aspect to our model of lag 3, as well as a Moving Average aspect with lag 3. This corresponds to an ARIMA(3, 3) model and will be one of the models tested in the next section.

Also based on the ACF and PACF, I think that there will be a seasonal aspect to my data as well due to how strong the seasonality is shown in Figure 4.

**Model Selection**

First I will run the auto.arima() function to see if my prediction for the model was close to what R deemed the best model. The results are shown below in Figure 6:

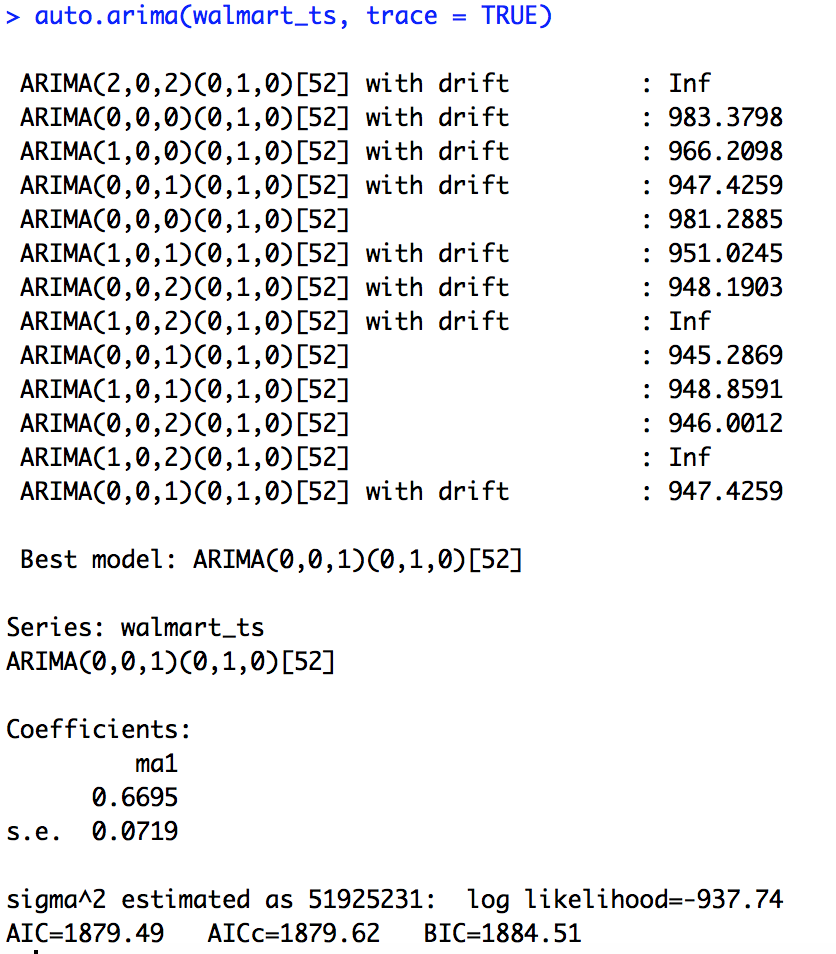


Figure 6: Base Model Generated by auto.arima()

Unfortunately my prediction was not correct (or even considered); the best model found by the software was an ARIMA(0, 0, 1)(0, 1, 0)[52].

The format for ARIMA is (p, d, q)(P, D, Q)[S]; the model above corresponds to an MA(1) model since the q value equals 1. There is also a seasonal difference of one (D = 1); the second set of numbers corresponds to seasonal Auto Regressive and Moving Average terms.

This would seem to suggest there is no AR aspect to the model since p = 0; however, I will test other values of p and q while keeping the seasonal part constant (D = 1).

But this is the base model, so I will be doing more testing for other models. Since the seasonal aspect of all of the models tested in Figure 6 was the same, I do not believe that the best models will have any differences in the seasonal part (though I tested for this anyway). Here is a table of models with their corresponding AIC values and whether or not all of their coefficient values were significant (i.e. if they were significant, they would not include 0 in their confidence intervals):

Note: The ACF and PACF of the differenced data was also obtained, and a possible model was included in the table below as Model 6 [the ACF/PACF graph is included in the appendix]

**Best Models**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | p | d | q | P | D | Q | AIC | All Sig? |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1879.62 | Yes |
| 2 | 1 | 0 | 1 | 0 | 1 | 0 | 1881 | No |
| 3 | 3 | 0 | 3 | 0 | 1 | 0 | 1868.16 | No |
| 4 | 1 | 0 | 2 | 0 | 1 | 0 | 1871.34 | Yes |
| 5 | 2 | 0 | 0 | 0 | 1 | 0 | 1869.41 | Yes |
| 6 | 3 | 0 | 2 | 0 | 1 | 0 | 1866.58 | No |
| 7 | 3 | 0 | 5 | 1 | 1 | 1 | 1868.2 | No\* |
| 8 | 3 | 0 | 2 | 1 | 1 | 1 | 1864.82 | No |
| 9 | 1 | 0 | 2 | 0 | 2 | 0 | 836.43 | No |
| 10 | 2 | 0 | 1 | 0 | 2 | 0 | 833.83 | Yes |
| 11 | 2 | 0 | 2 | 0 | 2 | 0 | 832.69 | No |
| 12 | 0 | 0 | 2 | 0 | 2 | 0 | 834.44 | Yes |

Table 1: Best Selected Models

Note: These models tested above had most to all of their coefficient values significant and ran into no errors when running the code in R. Also all models have a period of 52 (weeks in a year).

\*This was a very tempting model to use, but one of the 8 terms was not significant.

I chose these models because some were similar to the model that I had derived from the ACF/PACF (ARMA(3, 3)) as well as the model that was generated from the auto.arima() function (ARMA(0, 0, 1)(0, 1, 0)). I tried some models with higher p/q values, but models with lower p/q values seemed to have all of their coefficient values significant more of the time. In order for the model to be considered significant, all of its coefficients for its terms must be significant (if their confidence intervals include zero, the term might not even exist).

AIC stands for Akaike Information Criterion, which is a method of choosing between competing models via goodness of fit and simplicity of the model (less model parameters the better). It is calculated as:

-2 \* ln(L) + 2 \* p

where L is the maximized value of the log-likelihood and p is the number of parameters in the model. This is how the values in Table 1 are calculated. The lower the AIC value the better.

I changed the seasonal part of the arima() code and realized it was giving me models with better AIC values (but not necessarily residual diagnostics). These are near the bottom of Table 1.

According to Table 1, these are the best models with significant coefficients (all seasonalities includes):

1. ARIMA(2, 0, 0)(0, 1, 0)[52]
2. ARIMA(2, 0, 1)(0, 2, 0)[52]
3. ARIMA(0, 0, 2)(0, 2, 0)[52]

**Estimates of Coefficients/Confidence Intervals**

Here are their corresponding coefficient values and standard errors (S.E.):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | AR1 | AR2 | AR3 | MA1 | MA2 |
| A | 0.6776 | -0.5253 | N/A | N/A | N/A |
| S.E. | 0.0884 | 0.0876 | N/A | N/A | N/A |
| B | 0.6883 | -0.5789 | N/A | 0.5779 | N/A |
| S.E. | 0.1422 | 0.1330 | N/A | 0.1389 | N/A |
| C | N/A | N/A | N/A | 1.4445 | 0.9321 |
| S.E. | N/A | N/A | N/A | 0.1441 | 0.2402 |

Table 2: Selected Models, Coefficients & Standard Errors

And here are their corresponding Confidence Intervals (form 🡪 [Coeff +- critical value \* S.E.]; assuming a normal distribution of the data our critical value is 1.96 for a 95% CI):

Model A

AR1: 0.6776 +- (1.96 \* 0.0884) 🡪 [0.504336, 0.850864]

AR2: -0.5253 +- (1.96 \* 0.0876) 🡪 [-0.696996, -0.353604]

Model B

AR1: 0.6883 +- (1.96 \* 0.1422) 🡪 [0.409588, 0.967012]

AR2: -0.5789 +- (1.96 \* 0.1330) 🡪 [-0.83958, -0.31822]

MA1: 0.5779 +- (1.96\* 0.1389) 🡪 [0.305656, 0.850144]

Model C:

MA1: 1.4445 +- (1.96 \* 0.1441) 🡪 [1.16206, 1.72694]

MA2: 0.9321 +- (1.96 \* 0.2402) 🡪 [0.461308, 1.40289]

As you can above, all of our possible models have coefficients that we can say with 95% confidence that they will be nonzero. This means that all of the models chosen here will be valid (as opposed to the ones with “No” marked in Table 1 under “All Sig?”).

To figure out which of our models is the best, I will run residual diagnostics on all three.

**Residual Diagnostics**

Using the tsdiag() function I will determine the best model:

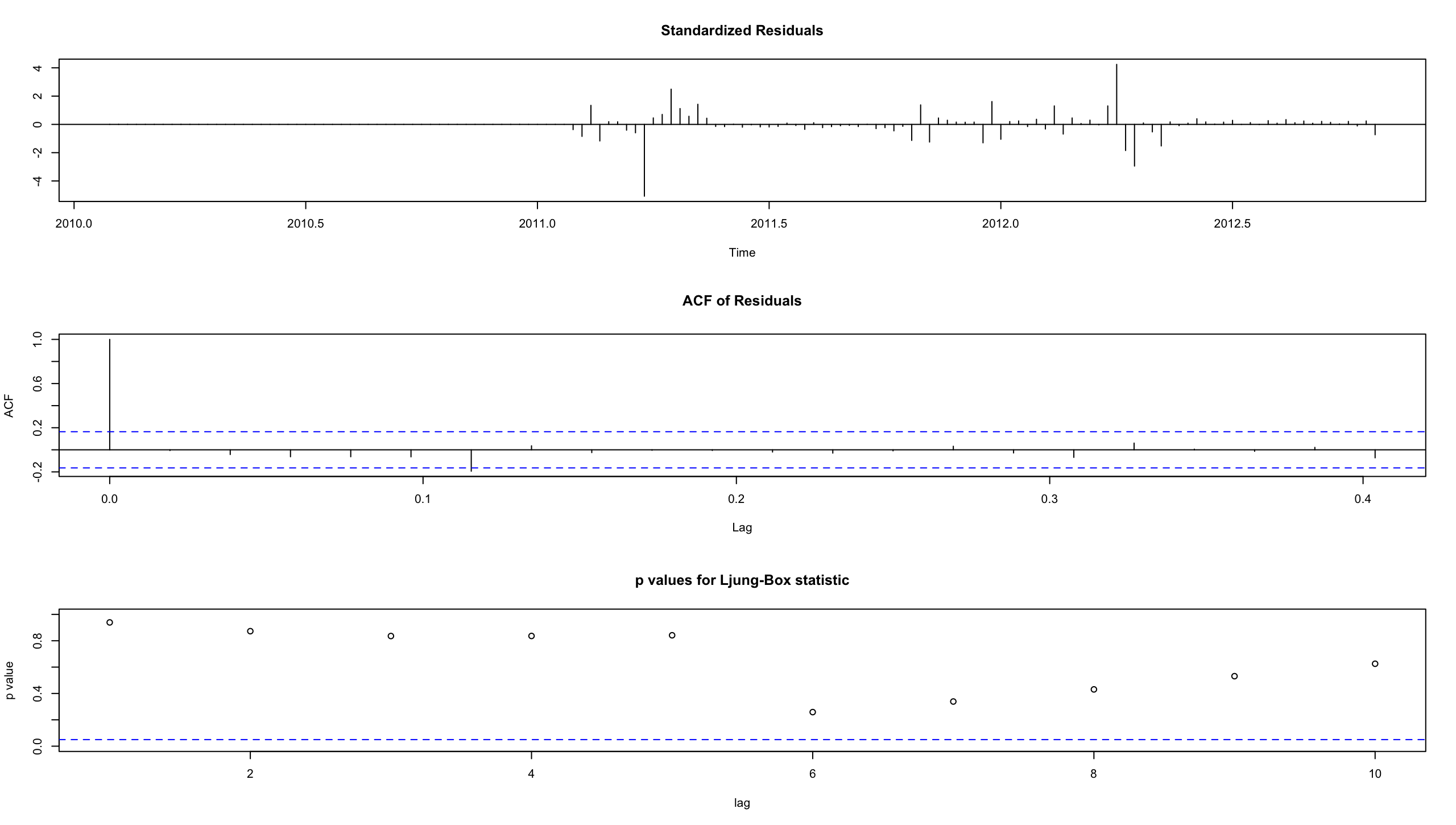


Figure 7: Model A Diagnostics

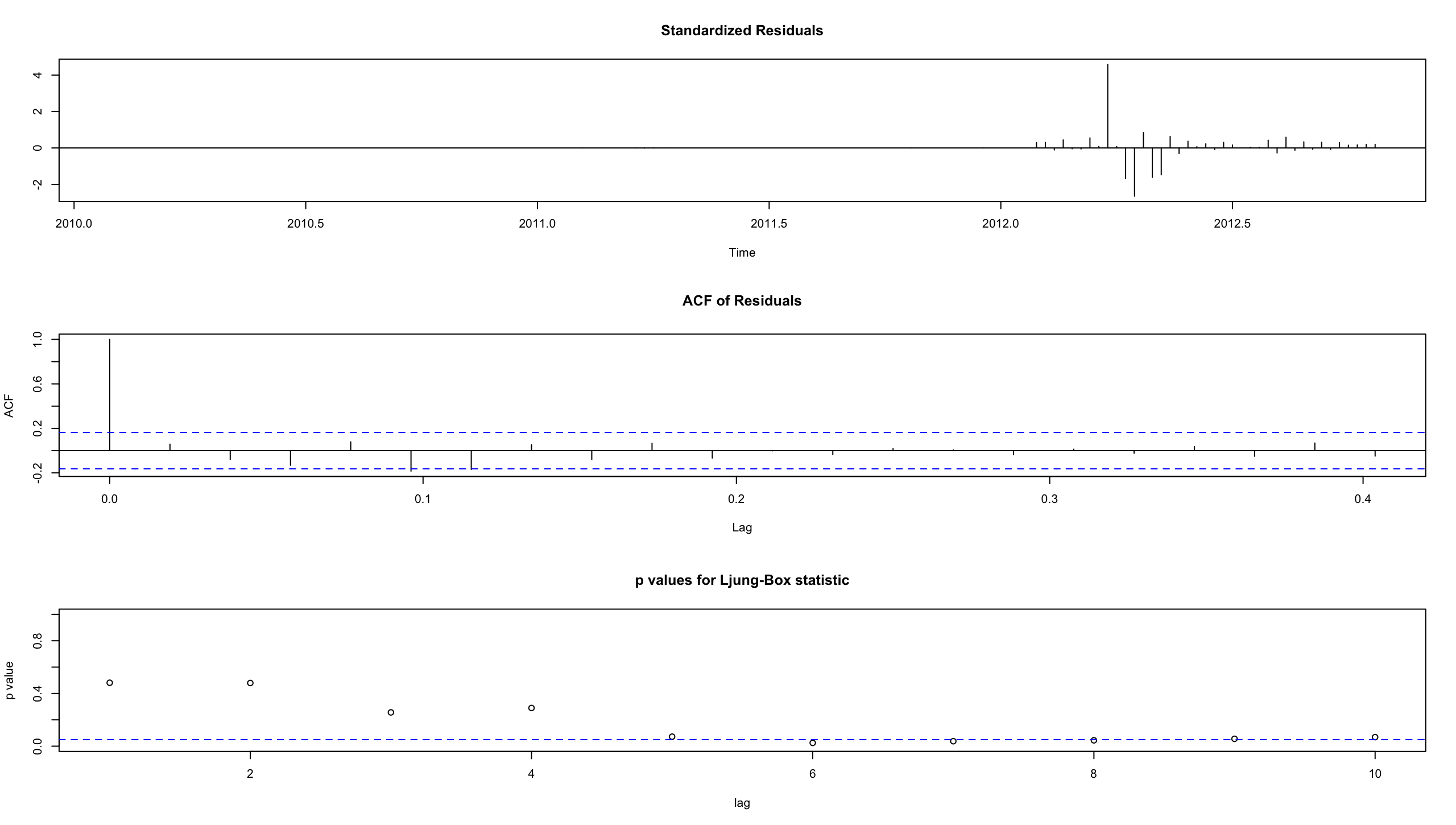


Figure 8: Model B Diagnostics

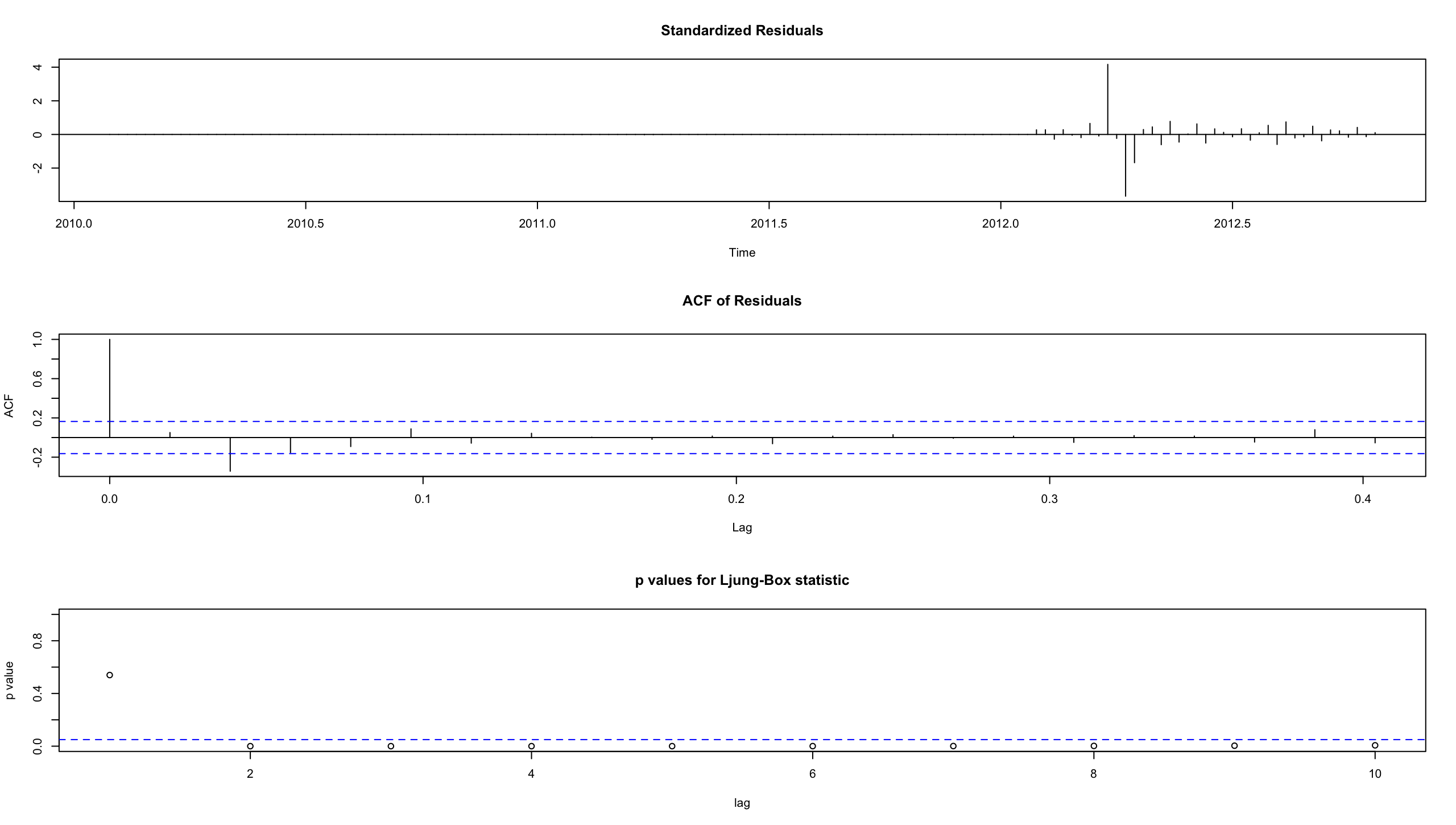


Figure 9: Model C Diagnostics

According to Figures 7, 8, and 9 above the standardized residuals seem random and uncorrelated, and the ACF has minimum lags outside the blue dotted lines.

Model B and C contain the lower amount of nonzero standardized residuals (flat residuals is not a good thing for model fitting), while Model A has the higher amount of nonzero standardized residuals (although the fact that some nonzero ones exist is a little concerning).

It also has the least amount of lag terms that are large (only the first one, which is always 1), compared to two for both Model B and C.

Therefore, Model A will be the model we use to forecast future values for Walmart sales. I will run a basic forecast as well as a 20 value forecast.

**Forecasting/Comparing**

To see if a forecast with this model is feasible, a general forecast using the forecast() function is shown here:

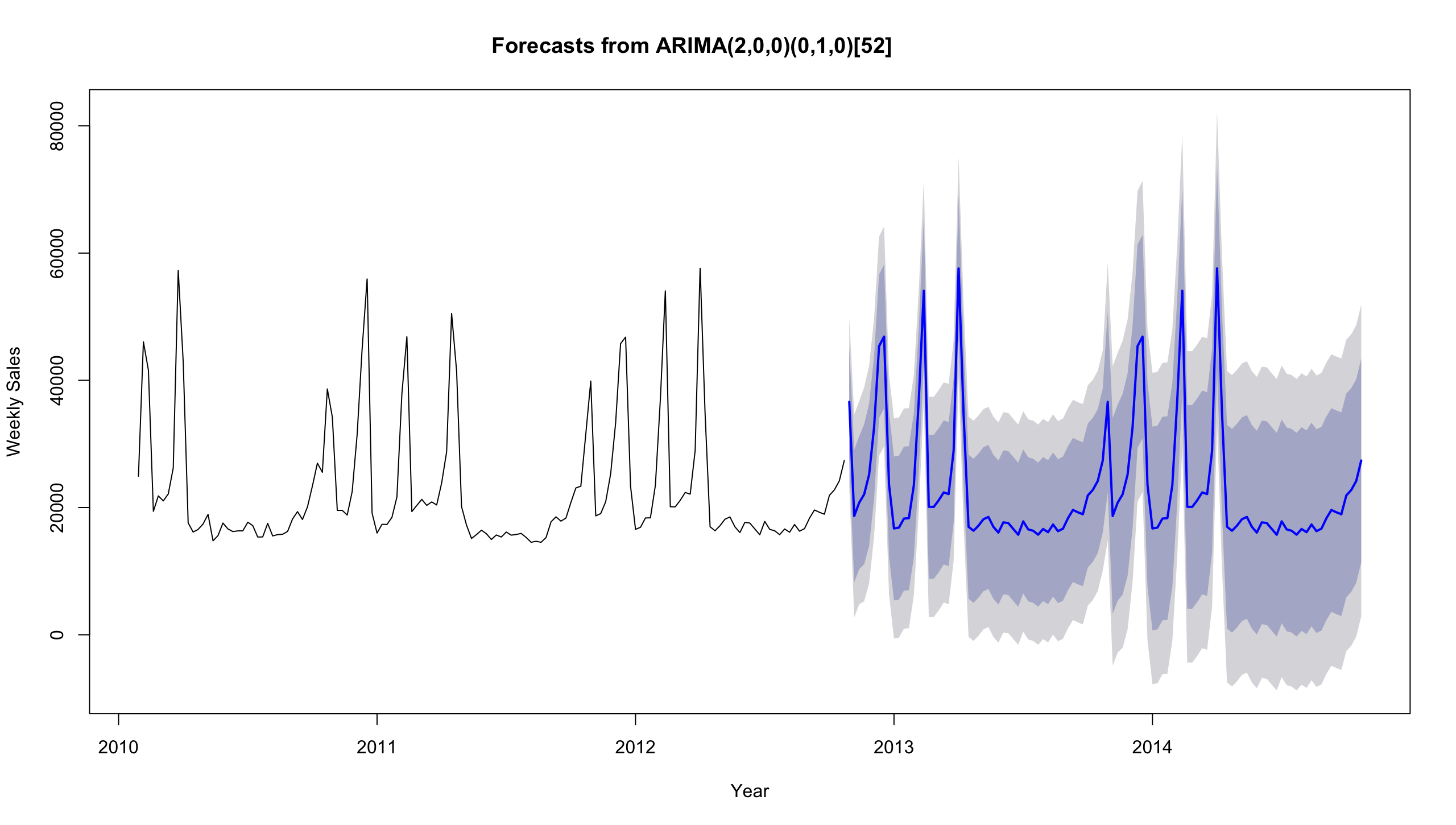


Figure 10: Initial Model Forecast

We see that there is a pretty consistent prediction interval for the next two years taking seasonality and trend into account well (even though that prediction interval includes 0 around 2014 and after, however unlikely that might be).

Now to really test our data: 10 of the last values will be removed from the original dataset, and those will be forecasted along with 10 future values. These predicted values will be graphed along with the actual values on the same plot to see how well the model can forecast values:

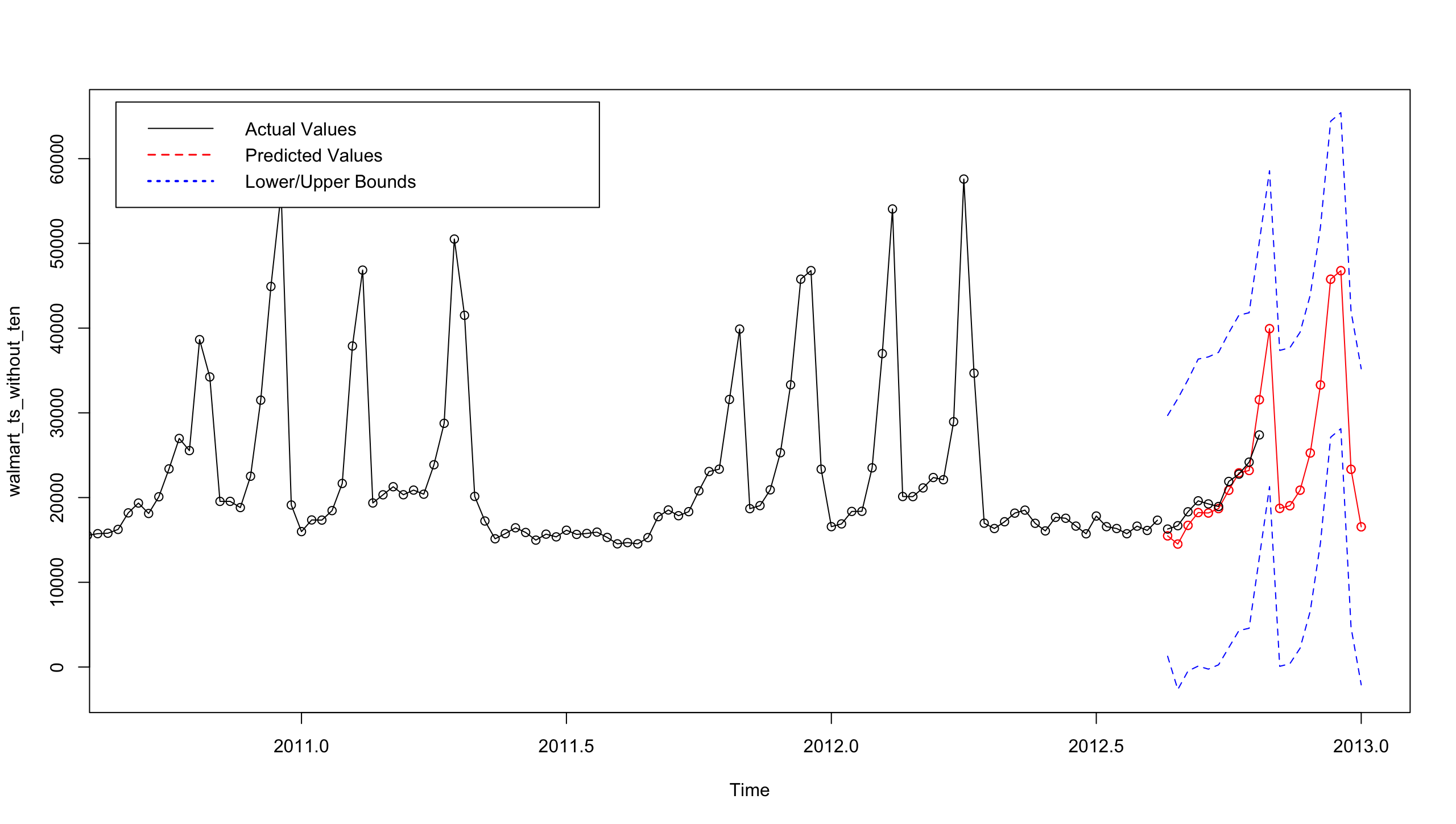


Figure 11: Forecasted Values vs Actual Values (Plus Future Values)

As we can see above in Figure 11, the predicted values come very close to the actual values, and seems to give a fairly reasonable prediction for the next ten values after the known data ends up until the end of 2013. Again, the standard error for this forecast must be relatively high considering that the lower bound of the confidence interval includes zero at a few points (and sales will never be zero barring some sort of crazy work or customer strike or other unlikely event) but the forecast does come close to the actual values and seems to capture the late year holiday rushes that most businesses experience well (the first spike is around Halloween since costumes are in high demand, and the second is around Christmas with no explanation needed).

**Conclusion**

While I was not able to get a model that minimized AIC while having significant coefficient values, the model I ended up with did a pretty good job forecasting values in the last part. Unfortunately, there is a good amount of standard error for each data point, leading to some of the prediction intervals containing zero as a possible value (no matter how unlikely that it) so I am sure that there are better options for models than this one. When I ran the forecast function() and the residual diagnostics on the decomposed data (after using the stl() function), this actually gave the best residuals out of all the models (no flat areas) which is why I included it in the list of potential models. However, since AIC for an ETS model is calculated differently than the AIC for an ARIMA model (initial values are treated differently), the AIC’s cannot be compared effectively. I was also very tempted to pursue with the differenced data set due to how well the residuals for it came out as well; however the forecast of this data using the model generated by auto.arima() gave a flat line for almost the whole forecast period, leading me to believe this method would not work. With so many possible models available for time series, it would be difficult to find the absolute best model in such a short period. In the end, someone needing a crude model to forecast Walmart’s weekly sales in this store and department probably could not do much worse than utilize this model. Perhaps to figure out times other than holidays where they can increase sales!

**References**

Data set: <https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting/data>

**Appendix**

Figure 1: Time Series Graph of Walmart Sales Data, Feb 2010 – October 2012: Used a frequency of 52 (weeks in a year) so the time series object can keep track of how often the patterns may repeat.

Figure 2: Decomposing the Original Data: This will separate the data into its seasonal, trend and random parts. The remaining residuals will usually indicate whether a transformation is needed.

Figure 3: Taking the First Difference of the Data: Taking the first difference of data means that every observation equals the difference of it and the next observation.

Figure 4: Decomposition of the Differenced Data: This was performed to see if the seasonality or the trend had lessened; the seasonality was still there and the trend had flipped.

Figure 5: Autocovariance function (ACF) & Partial Autocovariance function (PACF) Graphs: These graphs will show how many AR and MA terms would be in an ideal ARIMA model depending on the number of lags outside the blue dotted lines.

Figure 6: Base Model Generated by auto.arima(): Using the auto.arima() function with trace = TRUE, it will show the models generated and pick the best model for the user.

Figure 7: Model A Diagnostics: Graph of standardized residuals (top) of Model A, as well as the Standardized ACF values (middle) and a test for whether the errors are uncorrelated or not (bottom).

Figure 8: Model B Diagnostics: Same as above for Model B

Figure 9: Model C Diagnostics: Same as above for Model C

Figure 10: Initial Model Forecast: A basic forecast ahead of approximately two years using the final model.

Figure 11: Forecasted Values vs Actual Values (Plus Future Values): Forecasted ahead using sarima.for() to get predictions for the last 10 values as well 10 future values. Compared to the actual values.

Table 1: Best Selected Models: The best 13 models based on AIC.

Table 2: Selected Models, Coefficients & Standard Errors: Table of Models with their coefficients and standard errors for their terms to calculate confidence intervals.

ACF/PACF of Differenced Data (mentioned in page 6):

