**Walmart Sales Time Series Analysis**

USC Marshall, DSO 522 Time Series Analysis

*Team 8: Freda Lin, Zihan Ling, Pizheng Zhang, Zhiyu Zhang*

Table of Contents

[Executive Summary 3](#_Toc120540045)

[Data Description 4](#_Toc120540046)

[Data Cleaning 7](#_Toc120540047)

[Modeling 7](#_Toc120540048)

[Naïve and Seasonal Forecast 7](#_Toc120540049)

[Moving Average (MA) 8](#_Toc120540050)

[Trailing Average Smoother 9](#_Toc120540051)

[Simple Exponential Smoothing (SES) 11](#_Toc120540052)

[TBATS Model 12](#_Toc120540053)

[ARIMA 13](#_Toc120540054)

[Multiple Linear Regression 18](#_Toc120540055)

[Conclusion 23](#_Toc120540056)

## Executive Summary

The objective of the project is to build an optimal time series model to forecast the store sales of Walmart, which supports the company to make data-driven decisions in finance, marketing, supply chain, etc. The inspiration of conducting the sales forecast on the Walmart dataset comes from the demand for accurate performance prediction of the traditional retailing industry with the rise of e-commerce platforms in recent years. The final model algorithm can be scaled up to detect fraud in real-world problems. The data source of this project is from Kaggle.com. It contains historical sales data from 45 different Walmart department stores between 2010-02-05 and 2012-11-01. This report contains descriptions of data, the data cleaning process, the mechanism to create candidate variables, model algorithms, results, conclusions, and an appendix. In deploying model algorithms and fitting data, Naïve and Seasonal Forecasting, Moving Average, Trailing Average Smoother, Simple Exponential Smoothing, and TBATS model, ARIMA, and Multiple Linear Regression were used to select the best models for this project.

The model selected for the final algorithm is Multiple Linear Regression which used Lag\_Temperature.ts, Lag\_Fuel.ts, Xmas.ts, Ny.ts, HolidayFlag.ts, Cpi.ts as the independent variable. The model has the highest adjusted R-square 55.59% which means the combination of the variables we chose are the best interpretation of the sales. And we have concluded that (Business insight conclusion)

## Data Description

The time series data was reported weekly as “weekly\_sales” and there is no missing value in the dataset. In addition to date and weekly sales, the dataset also includes some relevant features such as whether it’s holiday season, temperature, fuel price, and consumer price index, which really helps us build a more robust model to forecast the store sales by taking these factors into consideration. The data also includes holiday events such as Super Bowl, Labor Day, Thanksgiving, and Christmas. There are 8 variables in the data set, 5 numerical variables and 3 categorical variables. We will illustrate each variable in the following:

A picture containing scatter chart

Description automatically generated

Numerical variables:

1. Weekly Sales: sales for the given store.

Chart, histogram

Description automatically generated

1. Temperature: temperature on the day of sale.

A picture containing chart

Description automatically generated

1. Fuel price: cost of the fuel in that region

Chart, line chart

Description automatically generated

1. CPI: Prevailing consumer price index

Chart, line chart

Description automatically generated

1. Unemployment: Prevailing unemployment rate

Chart, line chart

Description automatically generated

Categorical variables:

1. Store: the number of stores. There are 45 stores, and each store has 143 data.

Histogram

Description automatically generated

1. Holiday\_flag: whether the week is a special holiday week. 1 present it is holiday week. 0 presenstb it is non-holiday week

Chart, histogram

Description automatically generated

1. Date: date of sale. Each date has 45 data which is the 45 stores.

A picture containing table

Description automatically generated

## Data Cleaning

The purpose of data cleaning is to remove duplicate information, fix existing errors, and provide data consistency. After a close look at the dataset, there is no missing data, but the dataset contains 45 different stores. Therefore, we take the average of sales, temperature, Fuel\_price, CPI, Unemployment rate, and Holiday\_flag for each day and create a new time series data set. The cleaned data is shown as below:Table

Description automatically generated

## Modeling

A total of 7 model algorithms have been tested. During the modeling process, we split training and validation set. Training set is from 2010-02-05 to 2012-02-03 (105 weeks) and validation set is from 2012-02-03 to 2012-10-26 (38 weeks). Each model algorithm went through multiple iterations with a different number of variables and/or hyperparameters for the prediction performance, and the final best model was selected after comparing the best models of each algorithm.

A list of detailed algorithm explanations is provided below.

### Naïve and Seasonal Forecast

In the Naïve forecast, we used value of the last observation to set as our predictions of weekly sales. In seasonal naïve forecast, we set each forecast to be equal to the last observed value from the same season (e.g., the same week of the previous year).

Chart, histogram

Description automatically generated

As graph showed above, we can see that seasonal forecast has a better fit of testing data set. Seasonal forecast has a RMSE of 45729.70 and MAPE of 3.22061, and naïve forecast has a RMSE of 49706.2and MAPE of 3.5042. Although RMSE of naïve forecast is lower than RMSE of seasonal forecast, the difference between RMSE of naïve training set and RMSE of naïve testing set is a lot bigger than difference between RMSE of seasonal training set and RMSE of seasonal testing set (as the table showed below). The possible reason for that is because there is a peak at the end of the year, but naïve forecast cannot forecast that peak well, but seasonal forecast does. Thus, seasonal forecast performed better between these two methods.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ME** | **RMSE** | **MAE** | **MPE** | **MAPE** | **ACF1** | **Theil’s U** |
| **Naïve Train** | -783.148 | 158487.6 | 80656.62 | -0.9266072 | 7.296378 | -0.4198431 | NA |
| **Naïve Test** | 19824.370 | 49706.2 | 37439.69 | 1.7175670 | 3.504200 | 0.1430901 | 0.8345637 |
| **Seasonal Train** | 307.3048 | 43443.57 | 30138.88 | 0.0792618 | **2.872153** | 0.27863076 | NA |
| **Seasonal Test** | 26345.209 | 45729.70 | 34220.41 | 2.4457054 | 3.220261 | 0.04639538 | 0.7853771 |

### Moving Average (MA)

In the moving average section, we tried to take an average of the data within a time window. We did trial and error to split the time window to 4, 6, 12, and 18 weeks to select the best time window aka order, for the moving average. In the graph illustrated below, we can see that Moving Average 6 (MA6) has a better fit on the *WeeklySales* training set compared the rest of the three graphs. As result, we would go with MA6 for deeper analysis. With sufficient time, we also analyzed MA 4, MA12, and MA18 to ensure that our choice of MA is right.

Chart

Description automatically generated

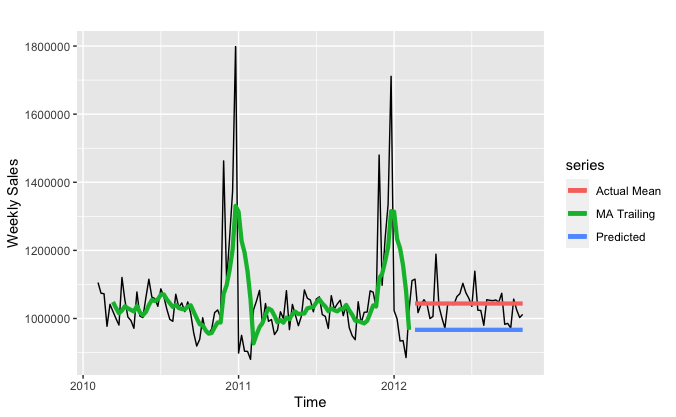
We further investigated MA 6 by applying **Trailing Moving Average**. After applied *rollmean* to the training data and set k=6, the graph is illustrated below:

Chart, histogram

Description automatically generated

### Trailing Average Smoother

In the trailing average smoother section, we used *k=6* and the last moving average for prediction. Since we are using last moving average to fill in all the 38 prediction weeks (nValid), the prediction period on the graphs shows to be a straight line illustrated below. This is a naïve way to predict Walmart’s weekly sales for the next 38 weeks since it only uses the last moving average.



To better make prediction for the following 38 weeks of store sales, we improved the trailing model by employing a for loop to append last moving average of not only the last moving average, but the rolling average for better prediction. We tested different time window: *k = 4, k = 6, k=12, and k=18* to see which time window would have a better fit and accuracy with the testing set.

Graphical user interface, chart, application, histogram

Description automatically generated

As graph showed below, we can see that graph *Trailing 4* and *Trailing 6* has better fit for the rolling line and test line; however, when we look at the accuracy score, we can see that trailing moving average with 12 weeks window has the lowest root mean square deviation （RMSE）and mean absolute percentage error (MAPE). As show in the table below, trailing moving average has a RMSE of 50593.8 and MAPE of 3.359, which is the lowest among the four.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ME** | **RMSE** | **MAE** | **MPE** | **MAPE** | **ACF1** | **Theil’s U** |
| **K = 4** | -3996.336 | 57580.03 | 42730.13 | -0.4816351 | 4.154352 | 0.2158959 | 2.842983 |
| **K = 6** | -5056.532 | 57088.12 | 42368.48 | -0.5701634 | 4.105455 | 0.23706 | 3.894264 |
| **K = 12** | 10326.97 | **50593.8** | 35462.27 | 0.9234055 | **3.359151** | 0.2333883 | 4.008283 |
| **K = 18** | 19367.91 | 56019.75 | 43931.72 | 1.754249 | 4.08718 | 0.2797184 | 5.85484 |

Conclusion: the best trailing moving average model is with time window of 12 weeks. It has RMSE of 50593.8 and MAPE of 3.359151.

### Simple Exponential Smoothing (SES)

Simple Exponential Smoothing is a popular forecasting method. It is similar from moving average, but instead of taking a simple average over a time window, it weights average of all past values, so that the weights decrease exponentially into the past. We are using seasonality as a testing stage to see how the data would react. We are using the artificial neural network (ANN) model to forecast the predicted store sales. We used lag=1 to look for the data for previous week.

Chart

Description automatically generated

With the ANN model accuracy, the RMSE is 60904.42 and MAPE is 191.3546.

Holt’s Linear Trend Model

When we put the data into Holt’s Linear Trend Model (ANN) to forecast 38 weeks of store sale, we can see in the graph that the predicted value is a line. Compared to the actual test set, the predicted line is not so accurate. Since exponential smoothing does not take seasonality into consideration, it is not very accurate. It has a RMSE of 46881.7 and MAPE of 3.284567.

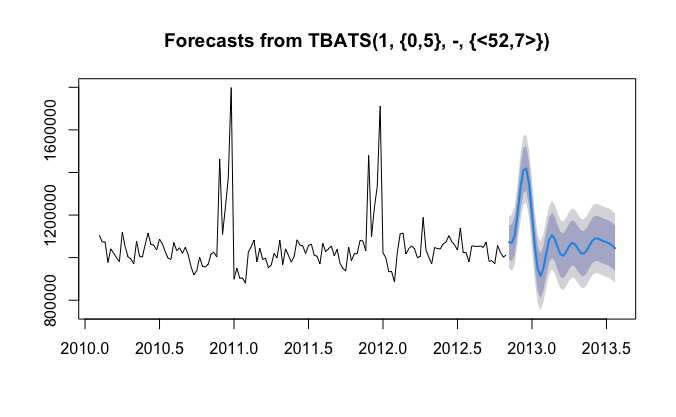
Chart, histogram

Description automatically generated

In conclusion for simple exponential smoothing, it is not a very good method to use to forecast or predict future values. Simple exponential smoothing does not take into consideration for trend and seasonality. In our graph, there are obvious seasonality at the end of the year of 2011 and 2012. So simple exponential smoothing is not a good predictor method for the Walmart dataset.

### TBATS Model

One trivial model we used is TBATS model. TBATS model has the capability to deal with complex seasonality with no seasonality constraints. TBATS is an acronym for key features of the model. **T**: Trigonometric seasonality, **B**: Box-Cox transformation, **A**: ARIMA errors, **T**: Trend, **S**: seasonal components.



As illustrated in the graph above, the forecast seems very reasonable. TBATS model takes it roots in exponential smoothing methods. In the graph, 1 is the Box-Cox parameter, and {0,5} is the ARMA (0,5) model. <52,7> is the seasonality length and Fourier series.

### ARIMA

In the ARIMA part, we consider both the ARIMA and Seasonal ARIMA models. One important assumption for both ARIMA and SARIMA is that the data needs to be stationary. To get rid of the trend component, we firstly compute the first difference of weekly sales. As the graph below shown, there is no obvious trend pattern left and there are some spikes and drops during Thanksgiving and Christmas each year. It seems that there are still some seasonal patterns remained in the data. However, because we only have about 2.5 years of time series data, we may not have enough data to train the model if we compute 52-week (or 1-year) difference of weekly sales. After consideration, we decide to use first difference data to train the models and we use Seasonal ARIMA to capture the seasonal patterns of the data.

Chart

Description automatically generated

Because we use first difference of weekly sales, the training set ranges from the 7th week of 2010 to the 6th week of 2012. We still regard the last 38 weeks of the dataset as the testing set. We check the ACF and PACF of the training set to determine what models should be applied.

Timeline

Description automatically generated with medium confidence

Table: ACF and PACF

Chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

From the above ACF and Partial ACF plots, we notice that the lag 52 is only significant for ACF and the correlation cuts off after the first season. For the first 51 weeks, or the first season, the ACF is cutting off and PACF is approximately tailing off. However, we also notice that the correlation becomes significant at the end of the first cycle. The most significant lags in ACF are lag(1), lag(4), and lag(5). According to the ACF and PACF table, we decide to train SARIMA(0,0,1)x(0,0,1)52, SARIMA(0,0,4)x(0,0,1)52, and SARIMA(0,0,5)x(0,0,1)52 and evaluate the performance of these models based on RMSE and MAPE in the training and testing set.

Model 1: SARIMA(0,0,1)x(0,0,1)52

Chart

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | RMSE | MAPE |
| Training set | 81944.13 | 387.72 |
| Testing set | 51303.22 | 216.95 |

Chart

Description automatically generated

Model 2: SARIMA(0,0,5)x(0,0,1)52

Chart

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | RMSE | MAPE |
| Training set | 69741.77 | 458.38 |
| Testing set | 52667.15 | 239.23 |

Graphical user interface, chart

Description automatically generated with medium confidence

Model 3: SARIMA(0,0,4)x(0,0,1)52

Chart

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | RMSE | MAPE |
| Training set | 76211.03 | 503.01 |
| Testing set | 49990.20 | 216.59 |

Graphical user interface

Description automatically generated

### Multiple Linear Regression

In multiple linear regression part, we introduce 7 extra variables. Five of them come from original dataset. Fuel price, Consumer Price Index (CPI) and Unemployment Rate capture the economic factors. Temperature and holiday flag capture demand factors. We also add two variables based on the time plot of sales. We notice that people tend to shop more from the beginning of thanksgiving week to the end of Christmas week, but tend to constraint consumption for several weeks around this period. Therefore, two dummy variables are introduced to capture those two special time periods seperately.

Table: Independent Variables in Multiple Linear Regression

|  |  |
| --- | --- |
| Variable Name | Meaning |
| Fuel.ts | Time series object of fuel price during the experimental period. |
| Cpi.ts | Time series object of Consumer Price Index during the experimental period. |
| Rate.ts | Time series object of Unemployment Rate during the experimental period. |
| Temp.ts | Time series object of temperature during the experimental period. |
| HolidayFlag.ts | Time series object of if national holiday during the experimental period. (Dummy) |
| Xmas.ts | Time series object of if Thanksgiving to Christmas Week during the experimental period. (Dummy) |
| Ny.ts | Time series object of if the weeks around Thanksgiving to Christmas Week during the experimental period. (Dummy) |

Next, we observe the relationship between each feature and sales, and find that: HolidayFlag.ts and Xmas.ts have positive relationship, Ny.ts has negative relationship, and other variables do not have obvious relationship. Then, we use Autocorrelation and Cross-Correlation Function Estimation (CCF) between each variable and sales to find lag relationship. Only Temp.ts and Fuel.ts have lag steps beyond the significance boundary.

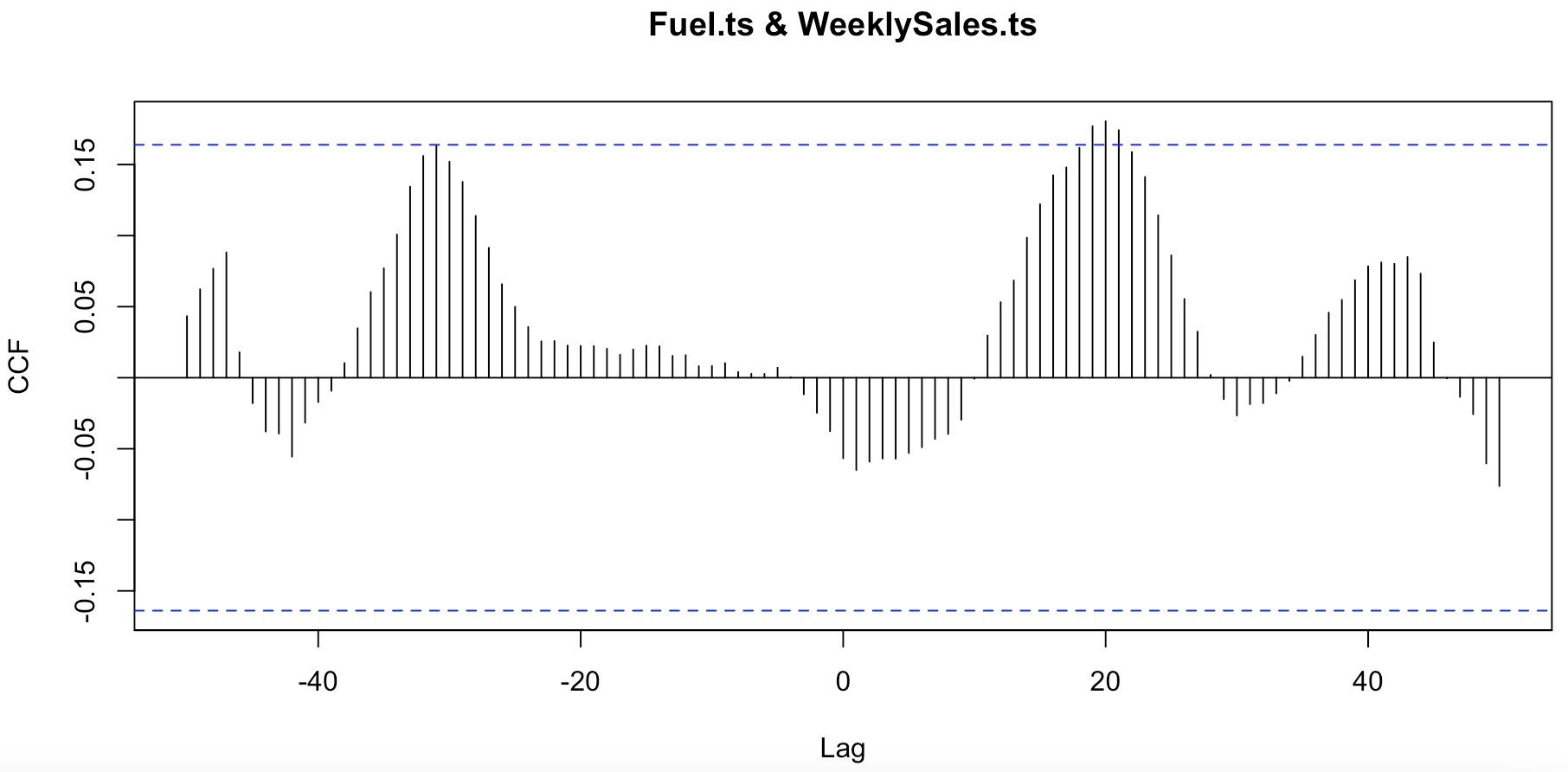


Figure: CCF Plot between Fuel Price and Weekly Sales Data

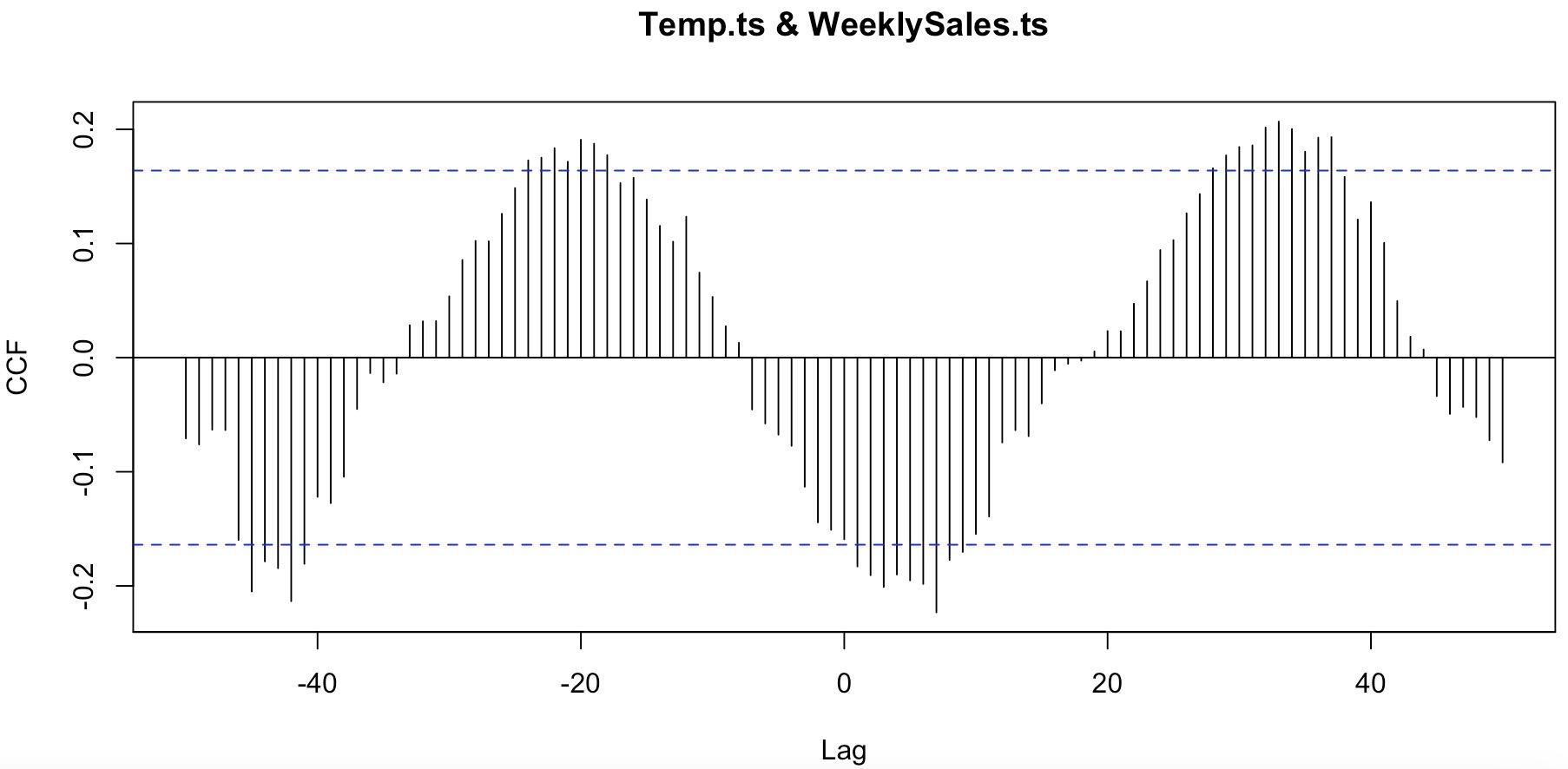


Figure: CCF Plot between Temperature and Weekly Sales Data

According to the CCF plots and simple linear regression function, we choose 20 steps for temperature and 31 steps for fuel price, and build new variables named Lag\_Temperatures.ts and Lag\_Fuel.ts correspondingly.

Finally, four multiple regression models with different combinations of variables are built to simulate the trend.

Table: Details of Multiple Linear Regressions

|  |  |
| --- | --- |
| Model | Independent Variables |
| m5 | Lag\_Temperature.ts, Lag\_Fuel.ts, Xmas.ts, Ny.ts |
| m6 | Lag\_Temperature.ts, Lag\_Fuel.ts, Xmas.ts, Ny.ts, HolidayFlag.ts |
| m7 | Lag\_Temperature.ts, Lag\_Fuel.ts, Xmas.ts, Ny.ts, HolidayFlag.ts, Cpi.ts |
| m8 | Lag\_Temperature.ts, Lag\_Fuel.ts, Xmas.ts, Ny.ts, HolidayFlag.ts, Cpi.ts, Rate.ts |

Each model’s performance is shown below.

Table: Performance of Multiple Linear Regressions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Adj-R2 | P-value | ME | RMSE | MAE | MPE | MAPE |
| m5 | 0.5537 | 0 | 0 | 88212.77 | 51726.3 | -0.53882 | 4.74 |
| m6 | 0.5537 | 0 | 0 | 87802.98 | 50790.72 | -0.53187 | 4.66 |
| **m7** | **0.5559** | **0** | **0** | **87167.58** | **51015.43** | **-0.52282** | **4.68** |
| m8 | 0.5518 | 0 | 0 | 87158.40 | 51103.75 | -0.52265 | 4.69 |

Considering the performance of four models, we can find that the model 7 has the highest adjusted R-square, which means that the combination of variables in model 7 can best interpret the variation in sales. At the same time, it has the approximate lowest loss among four models (adding another variable in model 8 will not dramatically reduce the error). Therefore, we choose m7 here. However, when we check the basic assumptions of model 7, we can obviously see the autocorrelation problem at lag 5. Therefore, we also add autocorrelation part into our original model, getting a new model named m7New.

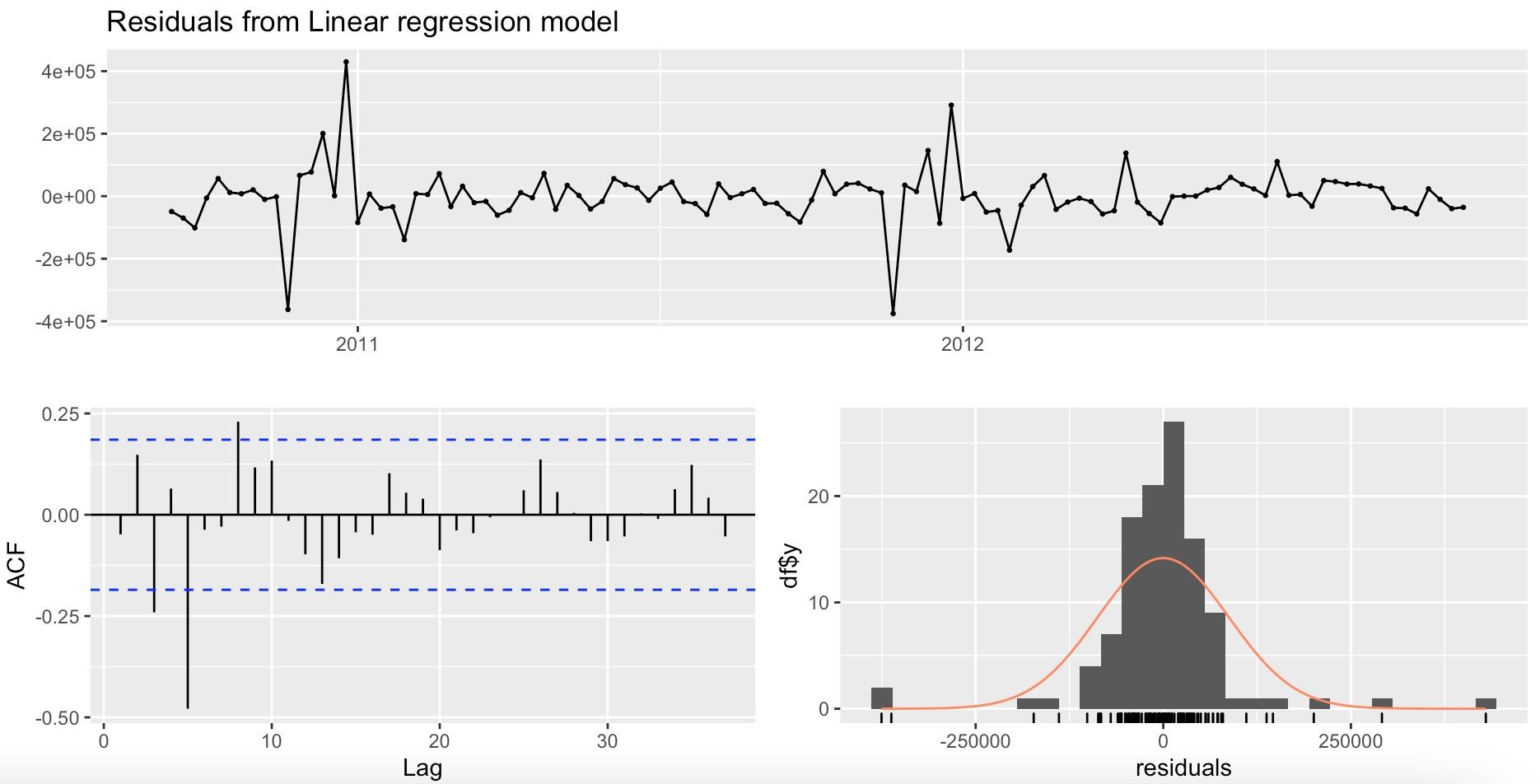


Figure: Check Model 7 Residuals

The Performance of final model will be:

Table: Performance of m7New

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | ME | RMSE | MAE | MPE | MAPE |
| m7New | -9.720363 | 69803.5 | 46924.75 | -0.056578 | 4.35 |

This model performs far better than the original model so this is our final choice. At last, we also use m7New to fit the original data.

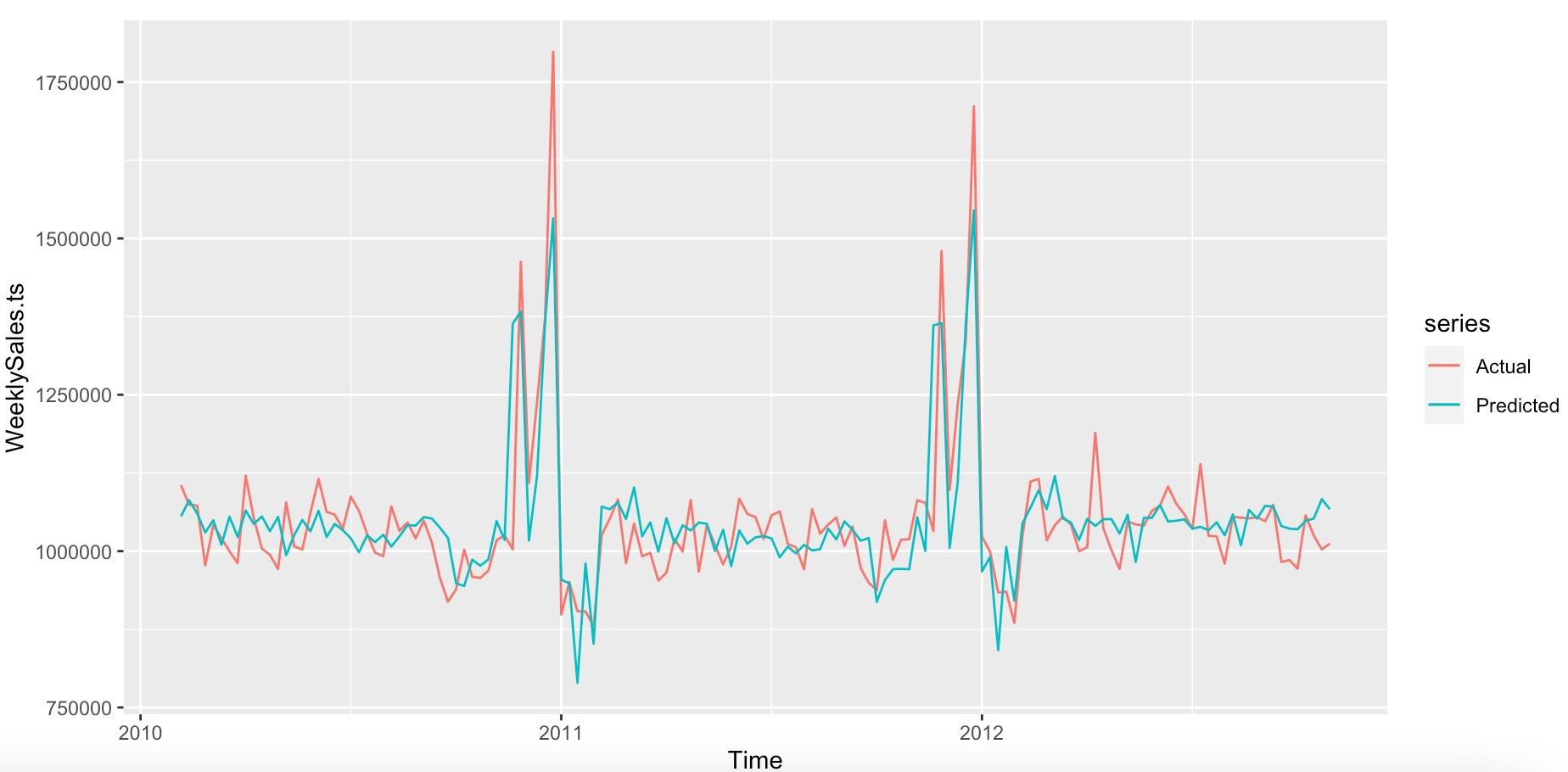


Figure: m7New Fitting Line

## Conclusion