Georgia State University | J. MACK ROBINSON COLLEGE OF BUSINESS

Mini Project: Predictive Analytics-MSA 8200

Weekly Sales Prediction – Walmart Time Series Data

Pranidhi Prabhat | Bennie Amani | Keerthi Bojja | Nimeelitha Akkiraju

March 15, 2020

Table of Contents

INTRODUCTION AND DATASET OVERVIEW

We were given a project on predicting sales using three different time series method.Our goal is to predict Weekly Sales of Walmart.

- Total Observations: 421,570
- Feb 2010 to Oct 2012: 143 weeks
- Stores: 45
- Departments: 90+
- Store Types: A, B & C

As the same can be done across a variety of parameters such as department, store or aggregated across Walmart. Our team chose to do the prediction for 3 stores with sales data aggregated across departments. We picked store number 3, 20 and 30 which are of store type B, A and C. Each store has the data of aggregated sales across department. The models can be extended to other stores, however one needs to be mindful about choosing the right parameters while running the regression by looking at the p-value of the coefficients in the regression model.

CSV Files from Kaggle:

Features: This file contains additional data related to the store, department, and regional activity for the given dates

- Store - the store number
- Date - the week
- Temperature - average temperature in the region
- Fuel_Price - cost of fuel in the region
- MarkDown1-5 - anonymized data related to promotional markdowns that Walmart is running. MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA.
- CPI - the consumer price index
- Unemployment - the unemployment rate
- IsHoliday - whether the week is a special holiday week

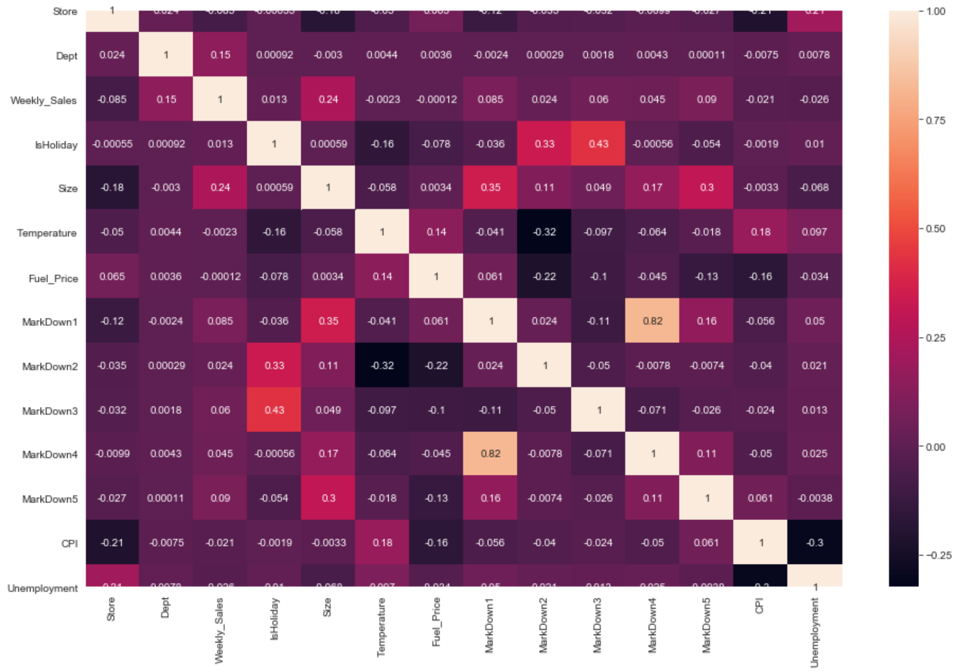
Stores: This file contains anonymized information about the 45 stores, indicating the type and size of store

Train: This is the historical training data

- Store - the store number
- Dept - the department number
- Date - the week
- Weekly_Sales - sales for the given department in the given store
- IsHoliday - whether the week is a special holiday week

EXPLORATORY DATA ANALYSIS

Correlation Matrix:



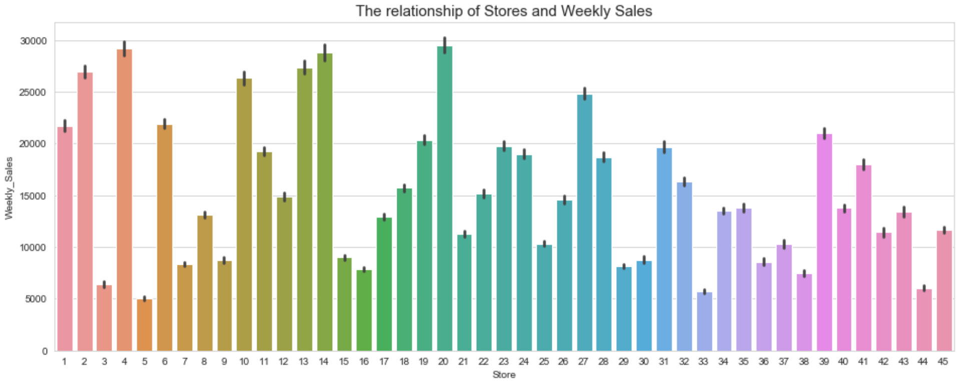
	Store	Dept	Weekly_Sales	IsHoliday	Size	Temperature	Fuel_Price	MarkDown1	MarkDown2	MarkDown3	MarkDown4	MarkDown5	CPI	Unemployment
Store	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Dept	0.024	1	0.15	0.00092	-0.003	0.0044	0.0036	-0.0024	0.00029	0.0018	0.0043	0.00011	-0.0075	0.0078
Weekly_Sales	-0.085	0.15	1	0.013	0.24	-0.0023	-0.00012	0.085	0.024	0.06	0.045	0.09	-0.021	-0.026
IsHoliday	-0.00055	0.00092	0.013	1	0.00059	-0.16	-0.078	-0.036	0.33	0.43	-0.00056	-0.054	-0.0019	0.01
Size	-0.18	-0.003	0.24	0.00059	1	-0.058	0.0034	0.35	0.11	0.049	0.17	0.3	-0.0033	-0.068
Temperature	-0.05	0.0044	-0.0023	-0.16	-0.058	1	0.14	-0.041	-0.32	-0.097	-0.064	-0.018	0.18	0.097
Fuel_Price	0.065	0.0036	-0.00012	-0.078	0.0034	0.14	1	0.061	-0.22	-0.1	-0.045	-0.13	-0.16	-0.034
MarkDown1	-0.12	-0.0024	0.085	-0.036	0.35	-0.041	0.061	1	0.024	-0.11	0.82	0.16	-0.056	0.05
MarkDown2	-0.035	0.00029	0.024	0.33	0.11	-0.32	-0.22	0.024	1	-0.05	-0.0078	-0.0074	-0.04	0.021
MarkDown3	-0.032	0.0018	0.06	0.43	0.049	-0.097	-0.1	-0.11	-0.05	1	-0.071	-0.026	-0.024	0.013
MarkDown4	-0.0099	0.0043	0.045	-0.00056	0.17	-0.064	-0.045	0.82	-0.0078	-0.071	1	0.11	-0.05	0.025
MarkDown5	-0.027	0.00011	0.09	-0.054	0.3	-0.018	-0.13	0.16	-0.0074	-0.026	0.11	1	0.061	-0.0038
CPI	-0.21	-0.0075	-0.021	-0.0019	-0.0033	0.18	-0.16	-0.056	-0.04	-0.024	-0.05	0.061	1	-0.3
Unemployment	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1

According to this heat map:

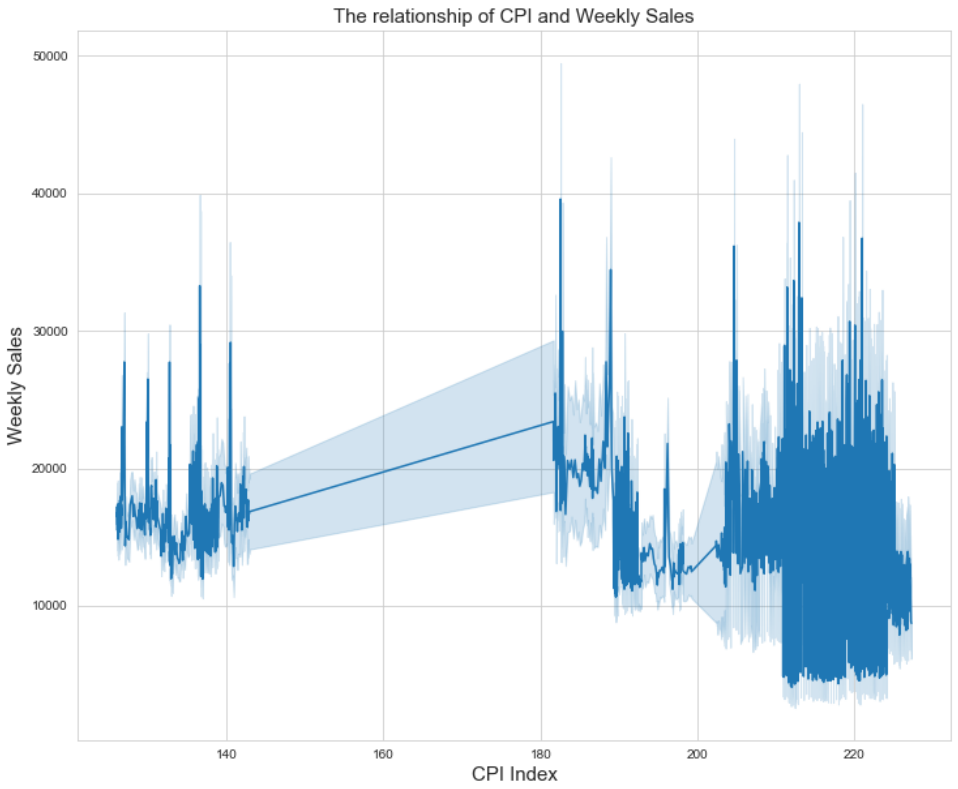
- None of the features have a very strong correlation with the Target Variable.
- MarkDown1 & MarkDown4 seem to be highly correlated with each other.

Further analysis & modeling is required to see how these features are contributing to the Weekly Sales

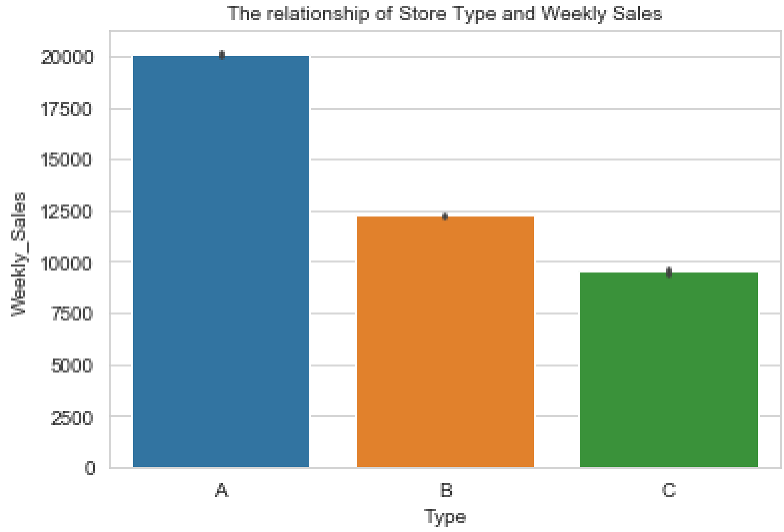
Relation between Stores & Weekly Sales:



Relation between CPI & Weekly Sales:

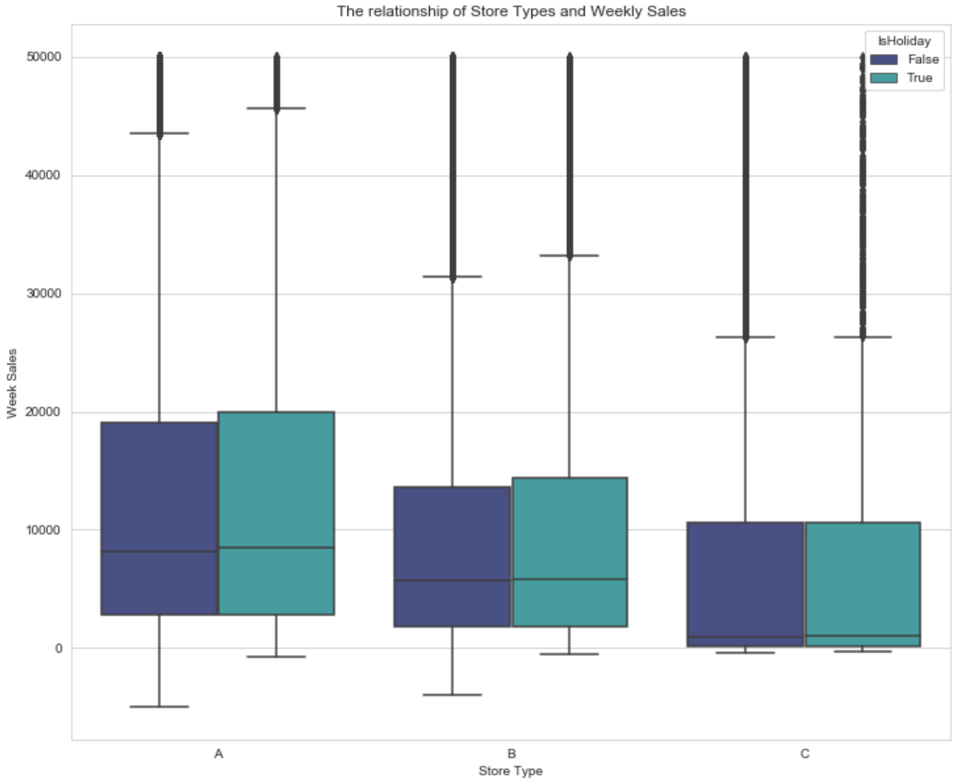


Relation between Store Type & Weekly Sales:



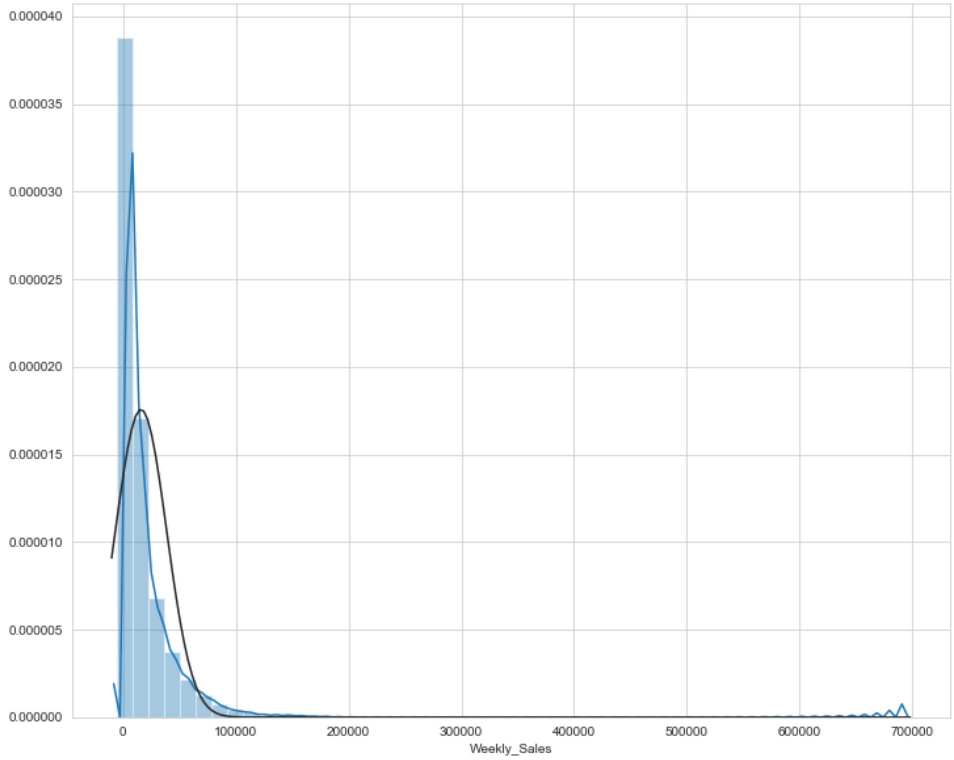
Store Type A has highest Weekly Sales followed by Type B and then Type C

Relation between Store Type, IsHoliday & Weekly Sales :



- Avg Sales: $15,981
- Variation in Weekly Sales is almost same within each Store Type irrespective of whether it is a Holiday week or not

Distribution of Weekly Sales:



DATA CLEANING & PRE-PROCESSING

- Merged the three files Train, Features & Store into a single data frame
- Filled null values with 0’s & deleted negative values in the target variable
- Aggregated weekly sales across department as there are more than 90 entries for each store
- Out of 45 stores, we randomly selected store no's 3, 20 & 30 from different store types for prediction
- Split the data into 80-20 for train and test

SARIMA with no additional variables

Seasonality and Model Selection

As mentioned above, we chose to randomly select one store in each type to simplify our analysis. The same methods were applied to each store.

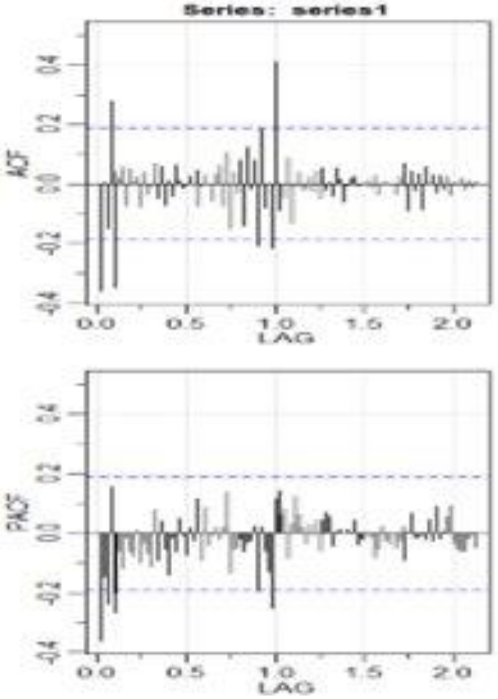
After dividing our data into a training and testing set, we plotted the ACF and PACF of the training set and use the function “decompose” to better understand the data.

We noticed a trend and variability, so we used the “diff” and “log” functions to make our data more stationary.

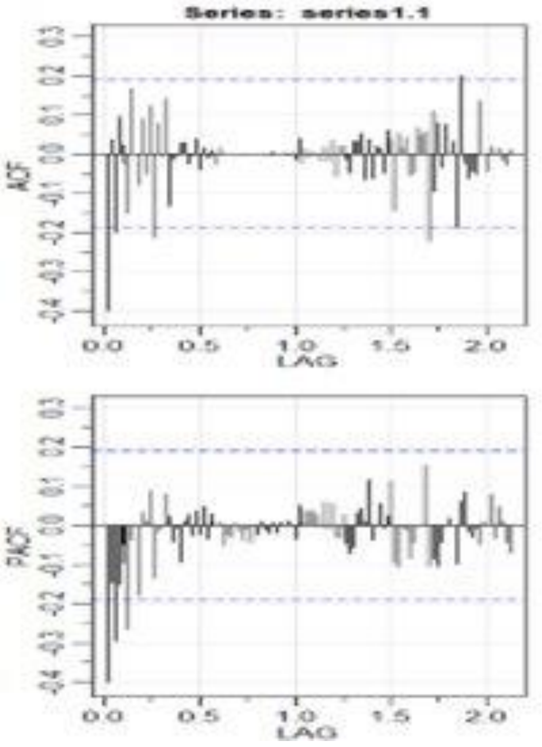
During our first attempt to use the SARIMA function on our transformed weekly data, we noticed that the seasonality component was not repeating faithfully every 52 periods; consequently, we decided to take the seasonality out of the data to make it easier to model.

Type A - Store 20

Below are the ACF and PACF of the transformed data with seasonality. We can observe that it is hard to assign a reliable seasonality component.

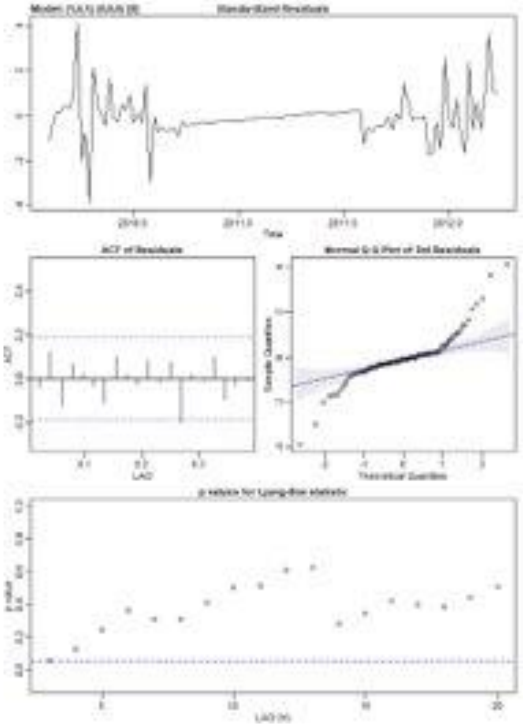


Below are the ACF and PACF of the transformed data without seasonality. We can see that the model we should use is ARMA(1,0,1).



SARIMA Model Output

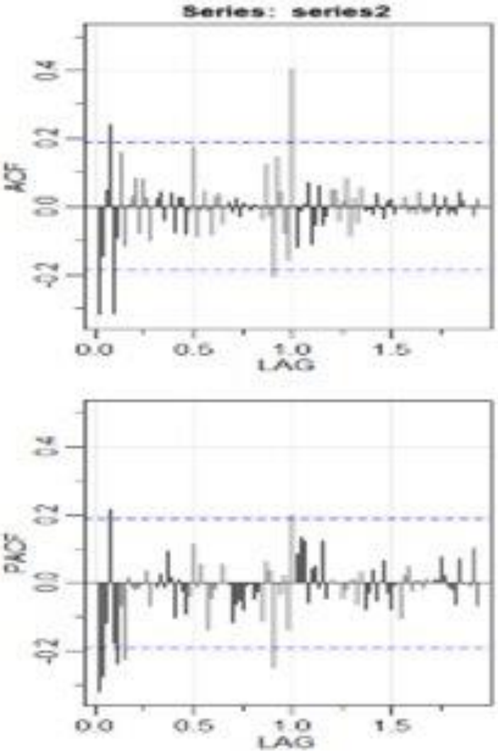
After running SARIMA(1,0,1,0,0,0,0) we get the following output:



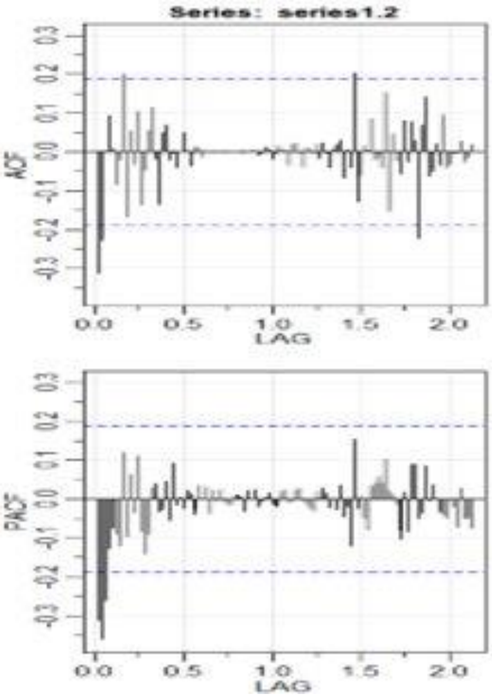
We are satisfied with this model because all the p-values are above the blue line.

Type B - Store 3

Below are the ACF and PACF of the transformed data with seasonality.

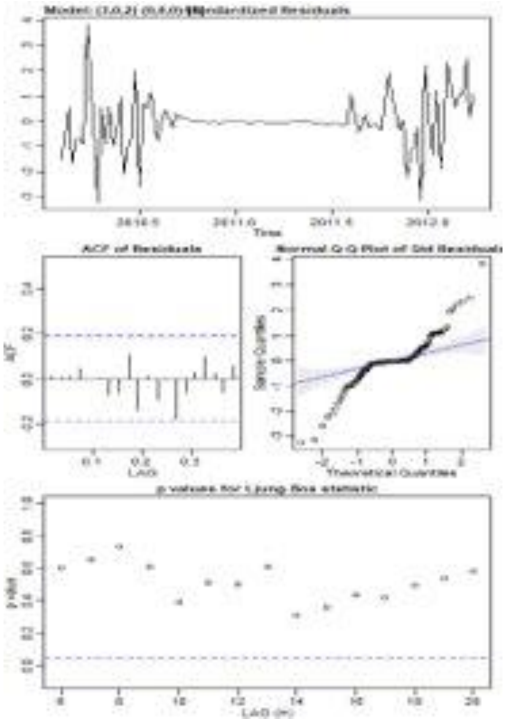


Below are the ACF and PACF of the transformed data without seasonality. Based on the plot, the model we chose to use is ARMA(3,0,2)



SARIMA Model Output

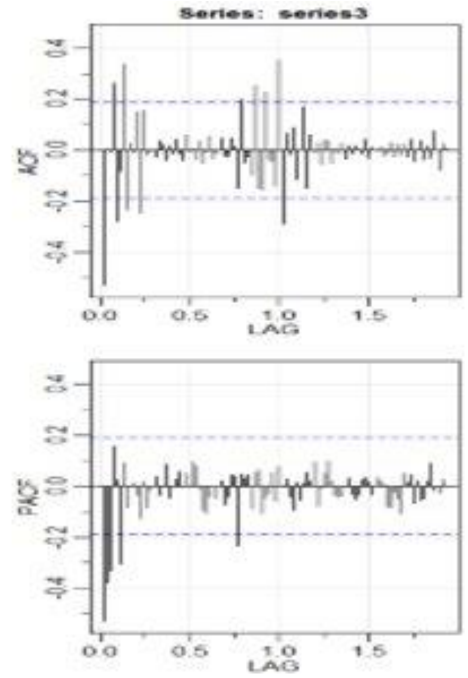
After running SARIMA(3,0,2,0,0,0,0) we get the following output:



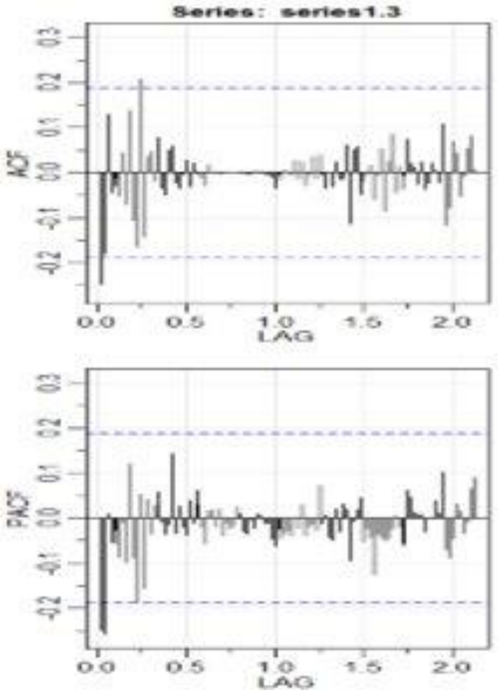
We are satisfied with this model because all the p-values are above the blue line.

Type C – Store 30

Below are the ACF and PACF of the transformed data seasonality.

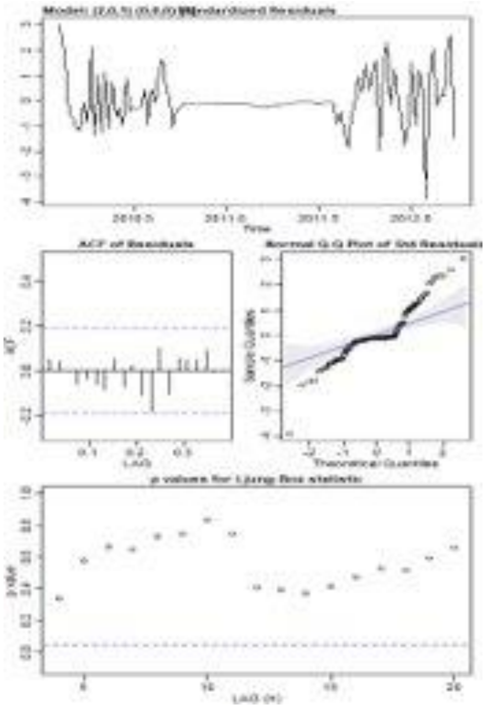


Below are the ACF and PACF of the transformed data without seasonality. Based on the plot, the model we chose to use is ARMA(2,0,1)



SARIMA Model Output

After running SARIMA(2,0,1,0,0,0,0) we get the following output:



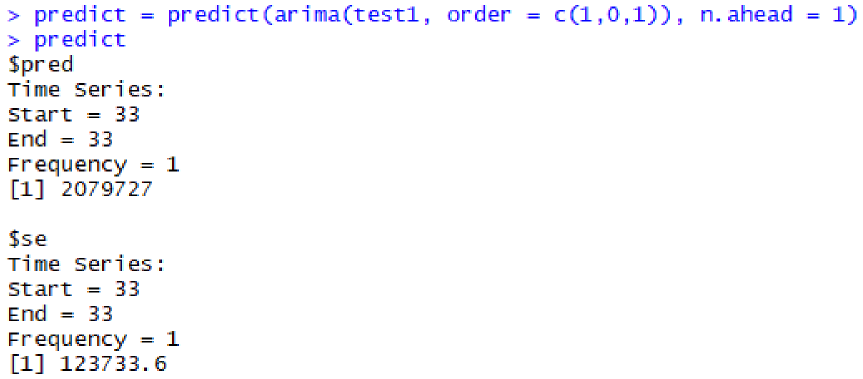
We are satisfied with this model because all the p-values are above the blue line.

Prediction and MAPE

We predicted the sales, calculated the errors and the Mean Absolute Percent Error (MAPE) for each of the selected stores.

Store 20:

Prediction:

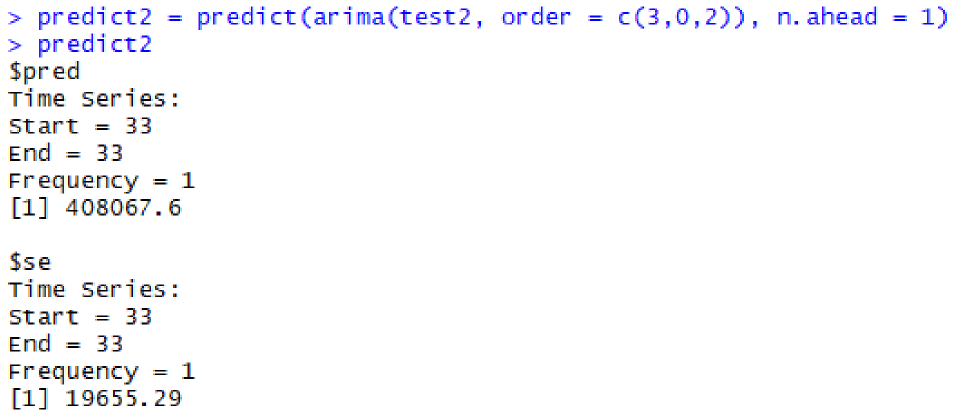


```
> predict = predict(arima(test1, order = c(1,0,1)), n.ahead = 1)
> predict
$pred
Time Series:
Start = 33
End = 33
Frequency = 1
[1] 2079727

$se
Time Series:
Start = 33
End = 33
Frequency = 1
[1] 123733.6
```

MAPE: 4.34%

Store 3:

Prediction:

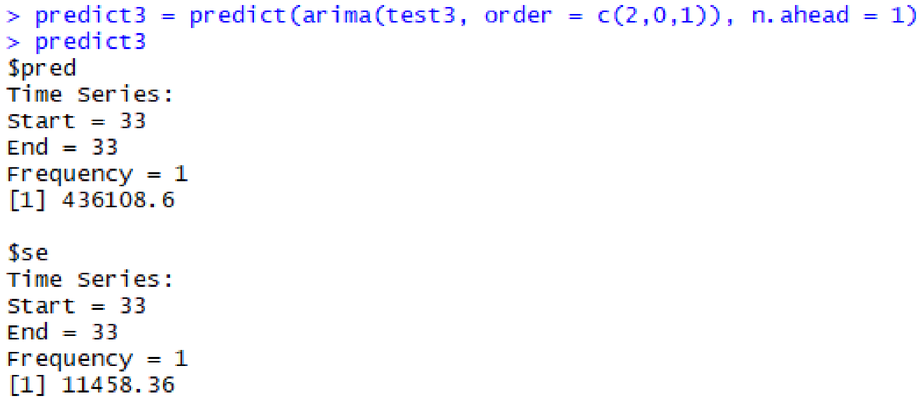


```
> predict2 = predict(arima(test2, order = c(3,0,2)), n.ahead = 1)
> predict2
$pred
Time Series:
Start = 33 
End = 33 
Frequency = 1 
[1] 408067.6

$se
Time Series:
Start = 33 
End = 33 
Frequency = 1 
[1] 19655.29
```

MAPE: 3.68%

Store 30:

Prediction:



```
> predict3 = predict(arima(test3, order = c(2,0,1)), n.ahead = 1)
> predict3
$pred
Time Series:
Start = 33 
End = 33 
Frequency = 1 
[1] 436108.6

$se
Time Series:
Start = 33 
End = 33 
Frequency = 1 
[1] 11458.36
```

MAPE: 2.24%

Comparison across 3 stores :

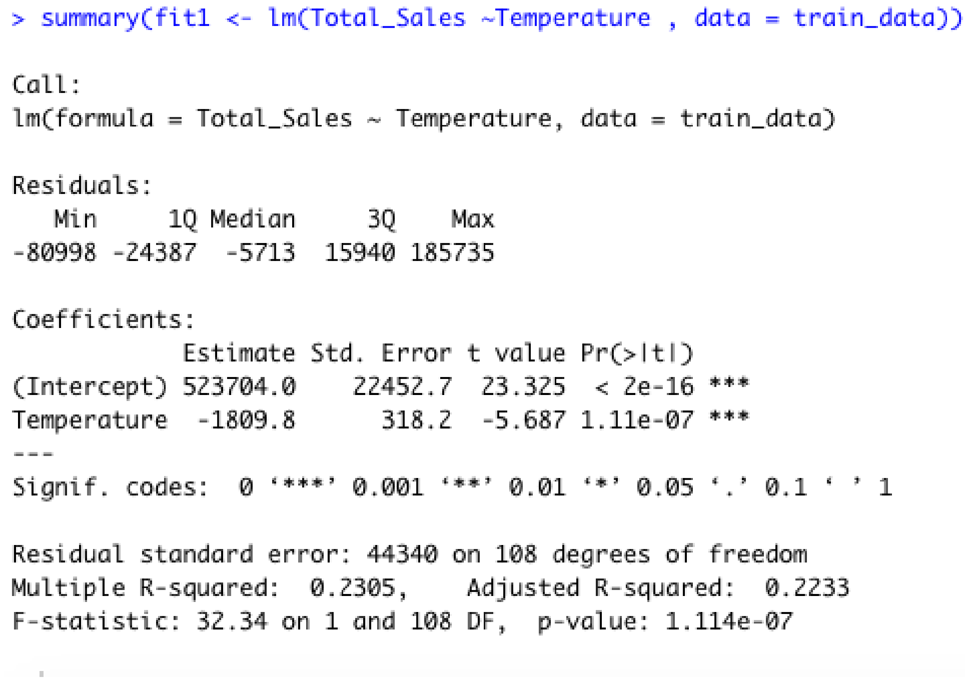
	Store3	Store 20	Store 30
MAPE	4.34	3.68	2.24

Overall, this method performed with strong accuracy

SARIMA with additional variables

Linear regression was ran for the three different types of store separately. All three stores yielded different models.

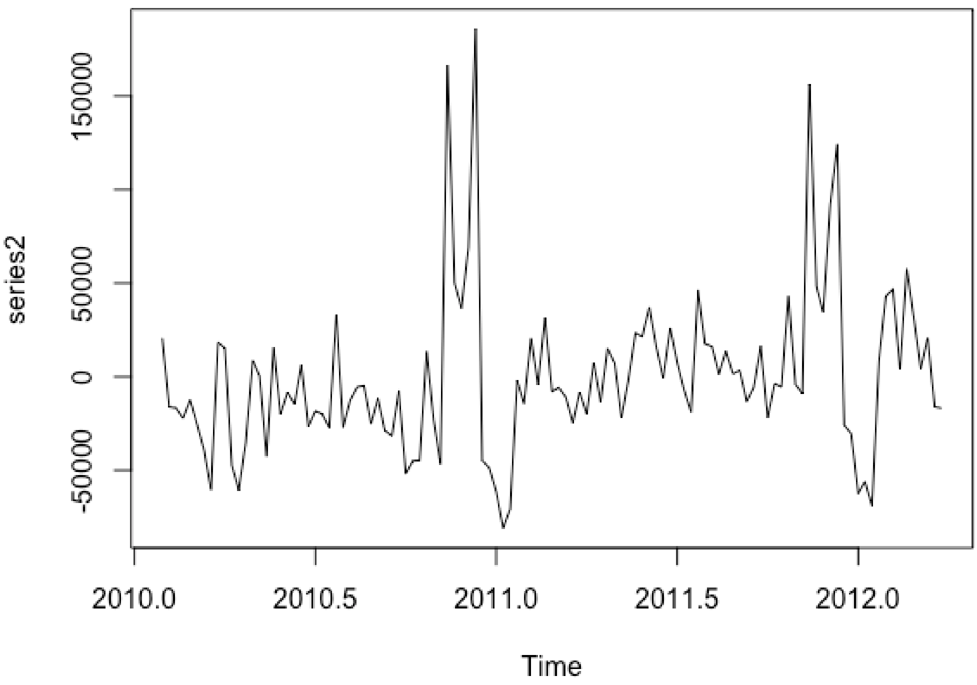
Store 3 :



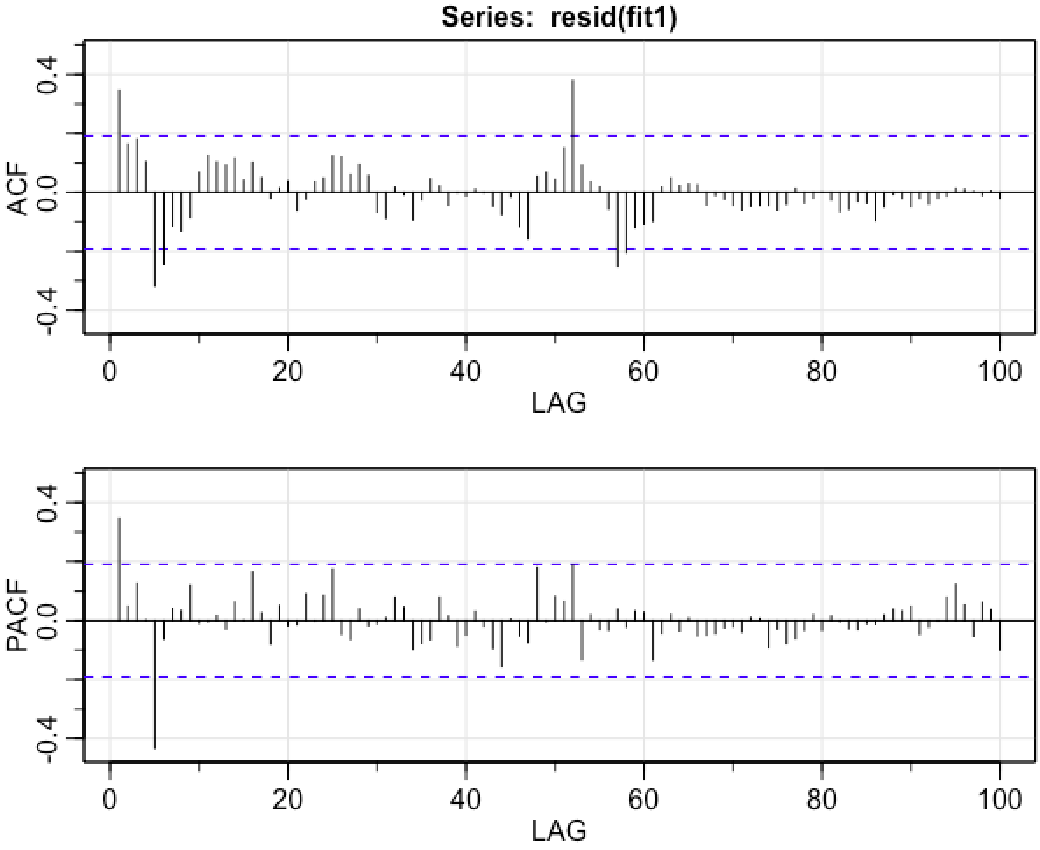
```
> summary(fit1 <- lm(Total_Sales ~Temperature , data = train_data))

Call:
lm(formula = Total_Sales ~ Temperature, data = train_data)

Residuals:
   Min     1Q Median     3Q    Max 
-80998 -24387  -5713  15940 185735 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 523704.0    22452.7  23.325  < 2e-16 ***
Temperature  -1809.8      318.2  -5.687 1.11e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 44340 on 108 degrees of freedom
Multiple R-squared:  0.2305,	Adjusted R-squared:  0.2233 
F-statistic: 32.34 on 1 and 108 DF,  p-value: 1.114e-07
```

As we see that the p-value for all the coefficients are smaller, this model was used for extracting the residuals.

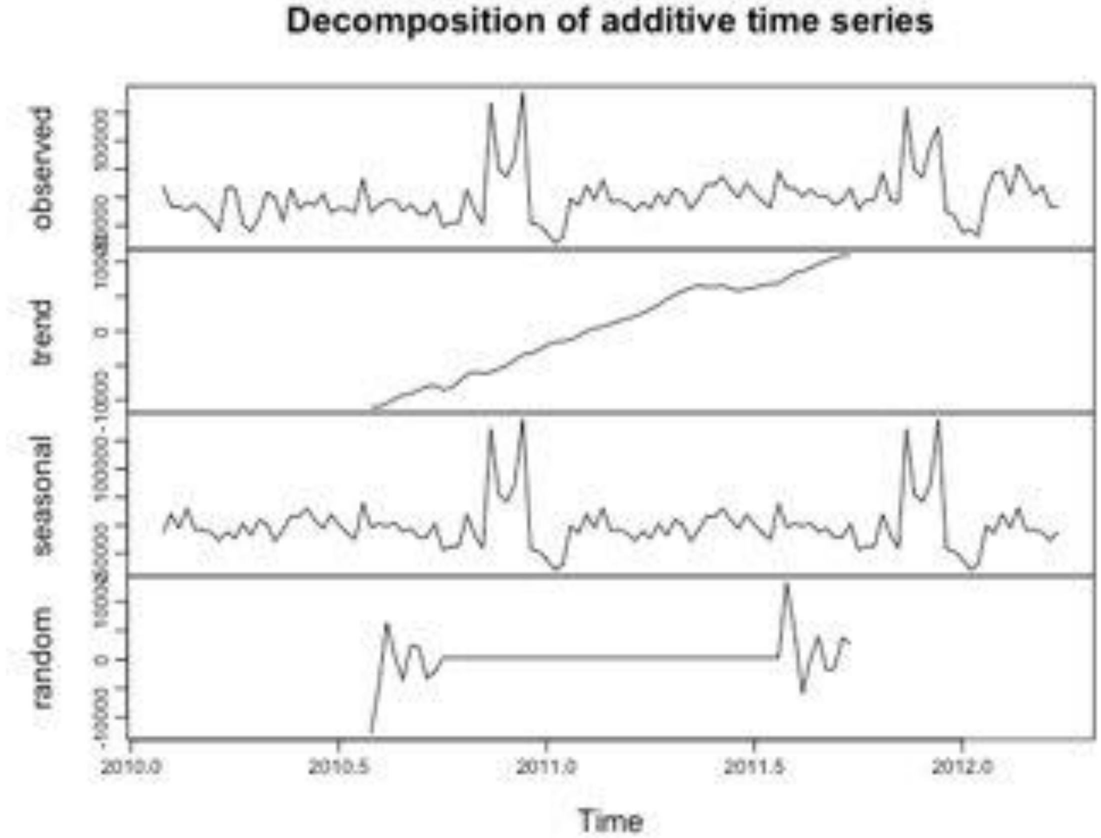
Following was the residual plot for the Store after running the regression model :



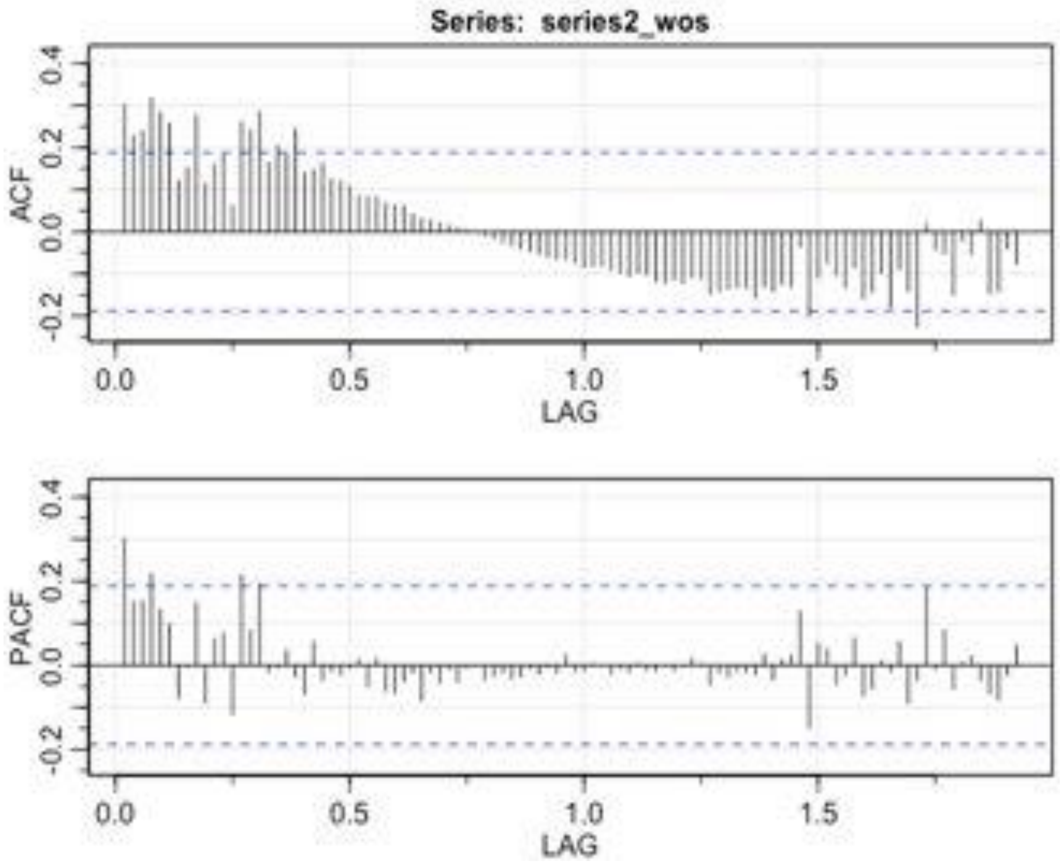
The acf plot for the residual looks like the following :



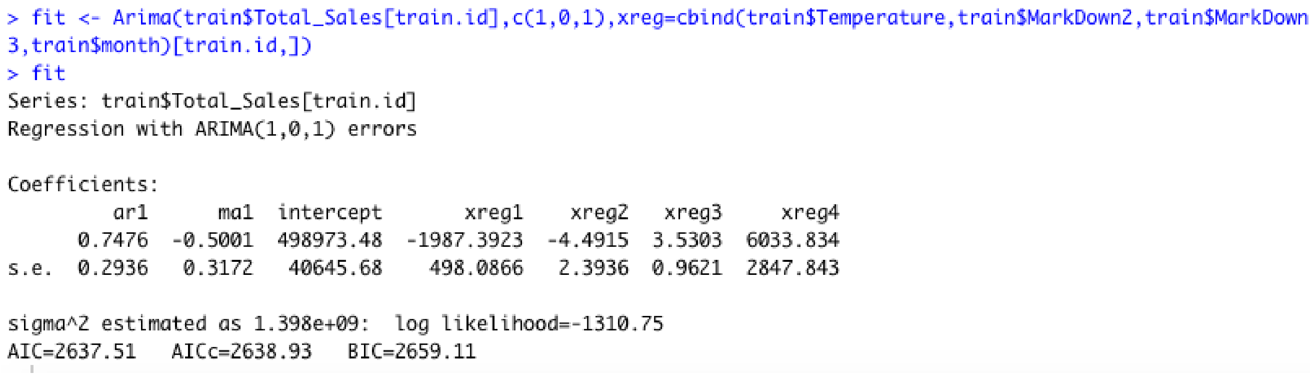
In the acf plot, there is a seasonality component at around 52 weeks, but as the data was limited to 143 weeks, it was not possible to use this seasonality component. So, we decided to remove the seasonality component by running the decomposition and subtracting the component.



After removing seasonality, below were the ACF plots :

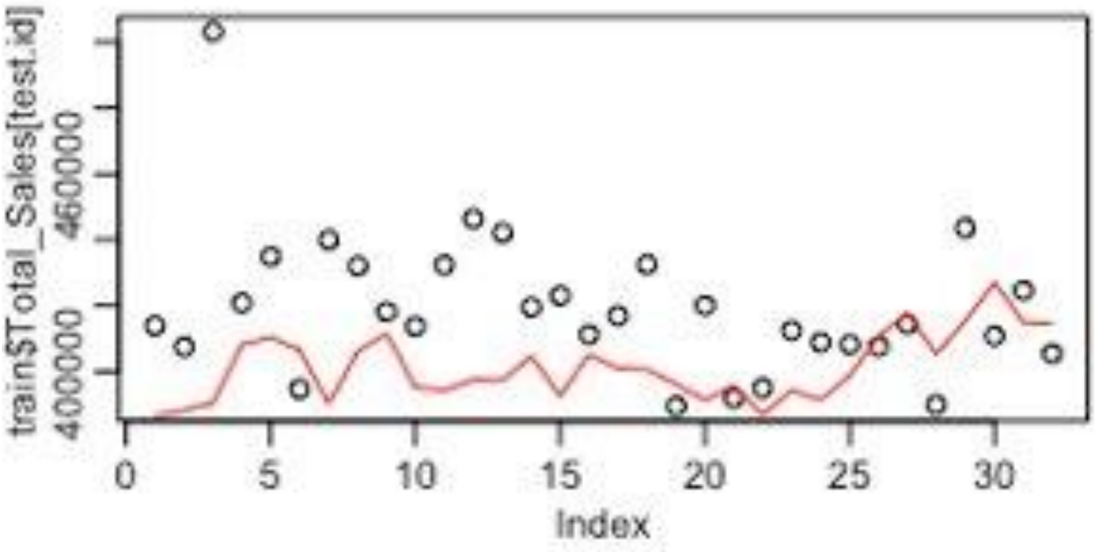


After examining the graph and trying different models, we selected AR3 MA3 process and ran the ARIMA function :

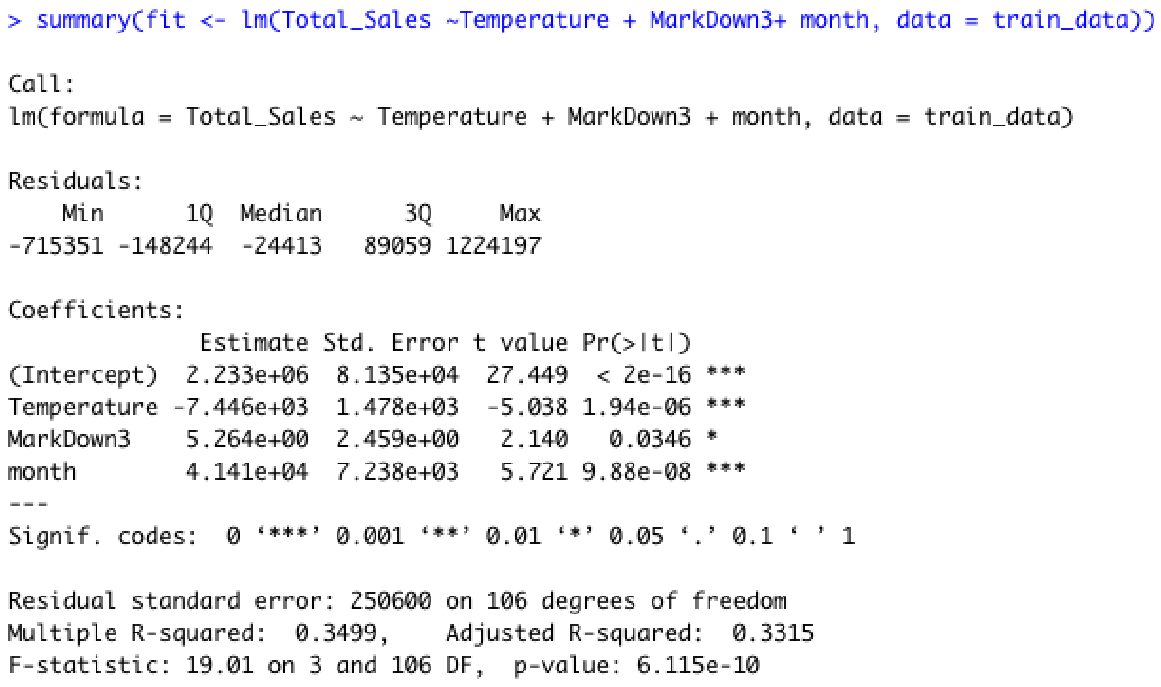


```
> fit <- Arima(train$Total_Sales[train.id],c(1,0,1),xreg=cbind(train$Temperature,train$MarkDown2,train$MarkDown
3,train$month)[train.id,])
> fit
Series: train$Total_Sales[train.id]
Regression with ARIMA(1,0,1) errors

Coefficients:
         ar1      ma1  intercept       xreg1    xreg2   xreg3     xreg4
      0.7476  -0.5001  498973.48  -1987.3923  -4.4915  3.5303  6033.834
s.e.  0.2936   0.3172   40645.68    498.0866   2.3936  0.9621  2847.843

sigma^2 estimated as 1.398e+09:  log likelihood=-1310.75
AIC=2637.51   AICc=2638.93   BIC=2659.11
```

The forecast result :



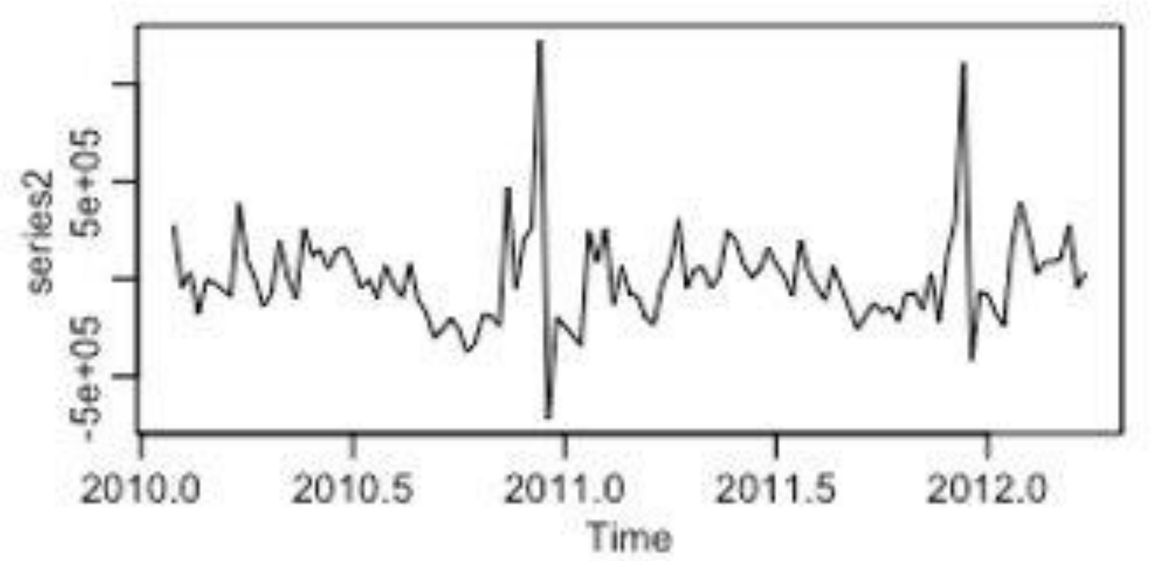
Store 20 :



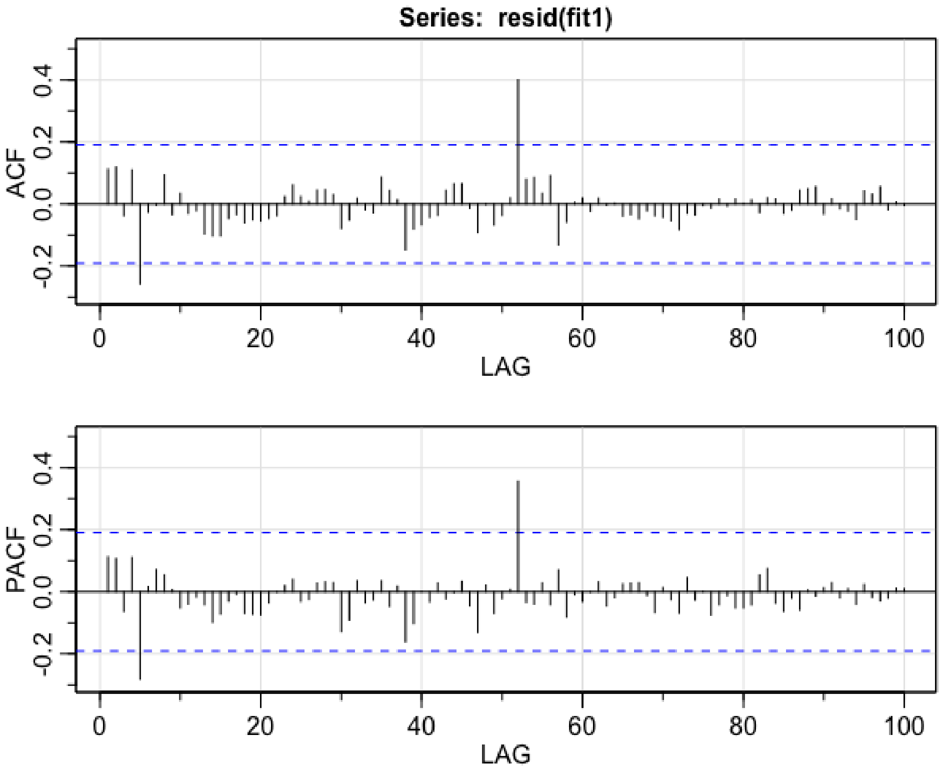
```
> summary(fit <- lm(Total_Sales ~Temperature + MarkDown3+ month, data = train_data))

Call:
lm(formula = Total_Sales ~ Temperature + MarkDown3 + month, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-715351 -148244  -24413   89059 1224197 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.233e+06  8.135e+04  27.449  < 2e-16 ***
Temperature -7.446e+03  1.478e+03  -5.038 1.94e-06 ***
MarkDown3    5.264e+00  2.459e+00   2.140   0.0346 *  
month        4.141e+04  7.238e+03   5.721 9.88e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 250600 on 106 degrees of freedom
Multiple R-squared:  0.3499,	Adjusted R-squared:  0.3315 
F-statistic: 19.01 on 3 and 106 DF,  p-value: 6.115e-10
```

As we see that the p-value for all the coefficients are smaller, this model was used for extracting the residuals.

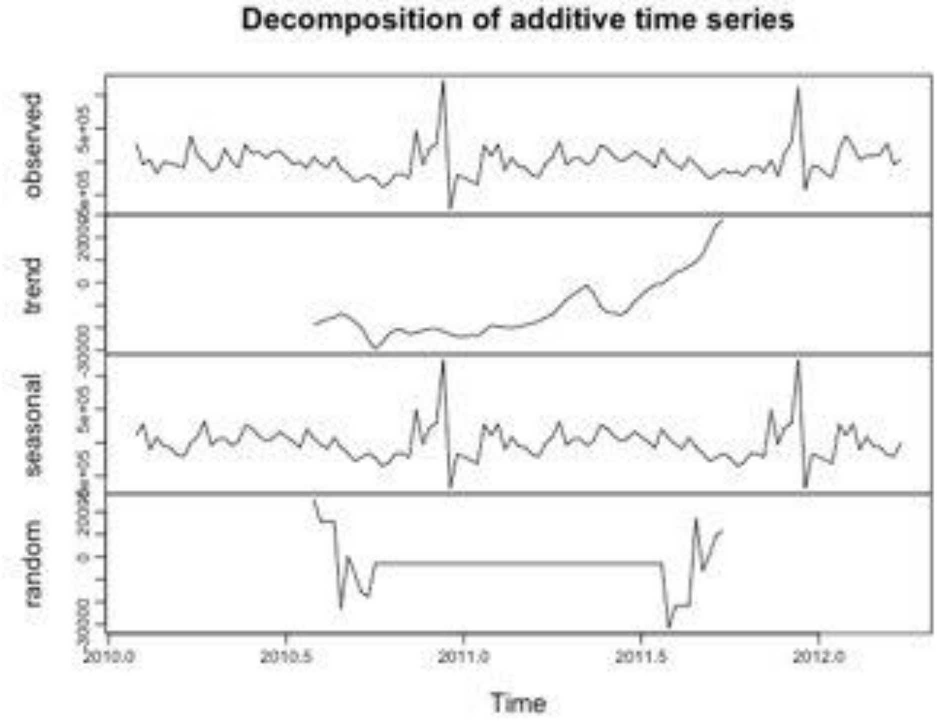
Following was the residual plot for the Store after running the regression model :



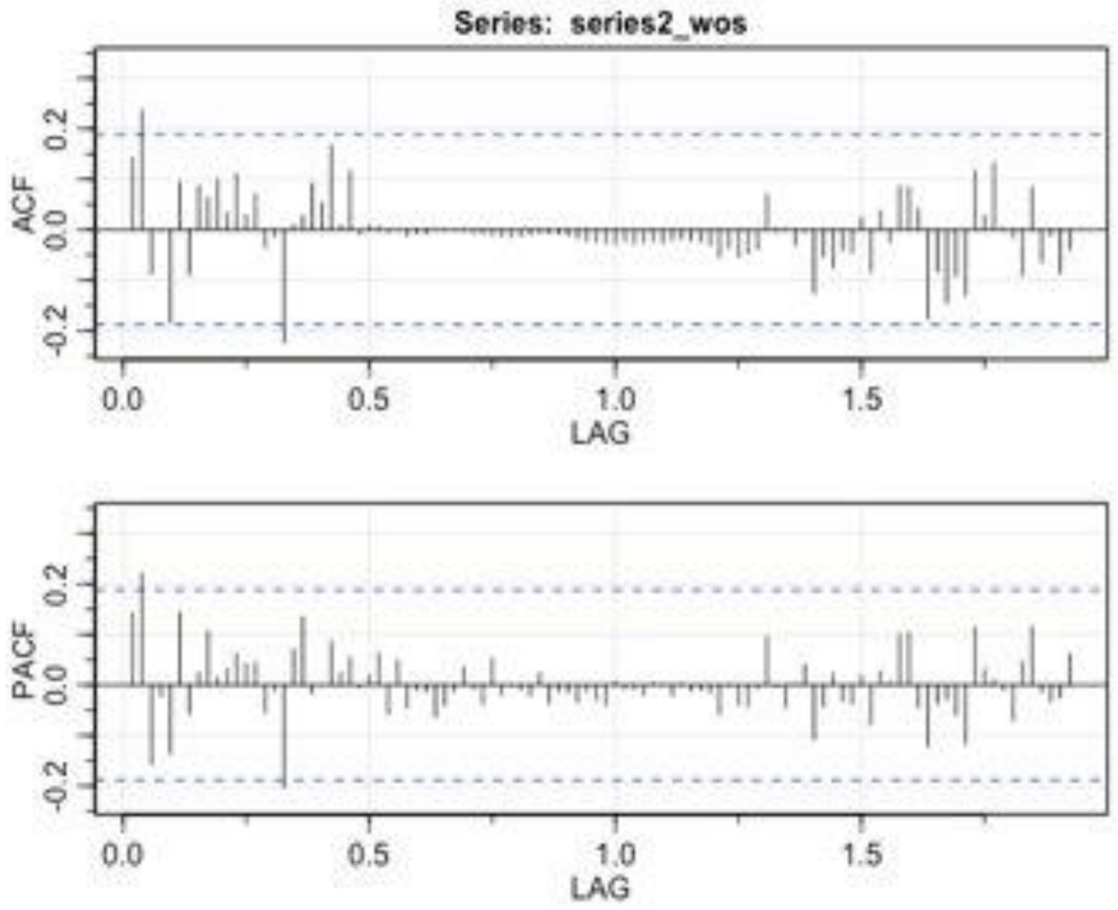
The acf plot for the residual looks like the following :



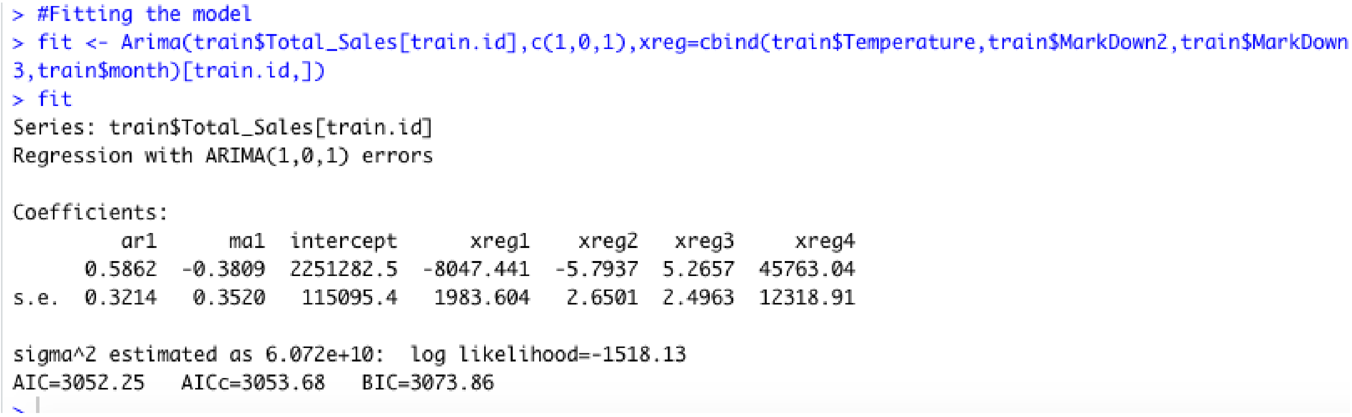
In the acf plot, there is a seasonality component at around 52 weeks, but as the data was limited to 143 weeks, it was not possible to use this seasonality component. So, we decided to remove the seasonality component by running the decomposition and subtracting the component.



After removing seasonality, below were the ACF plots :

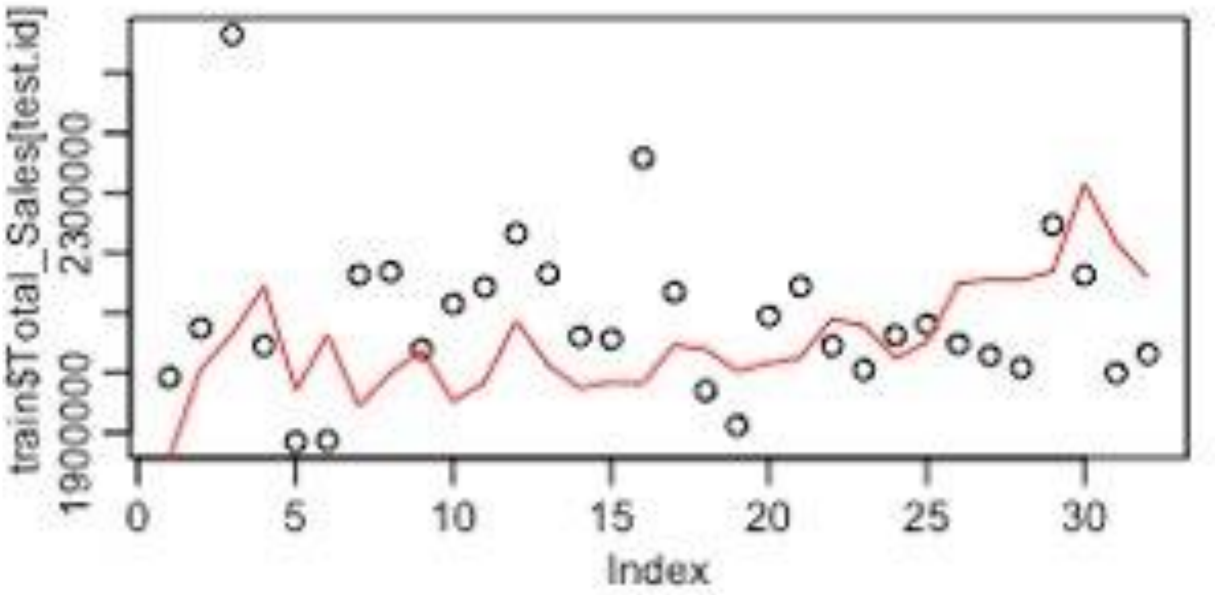


After examining the graph, we selected AR1 MA1 process and ran the ARIMA function :

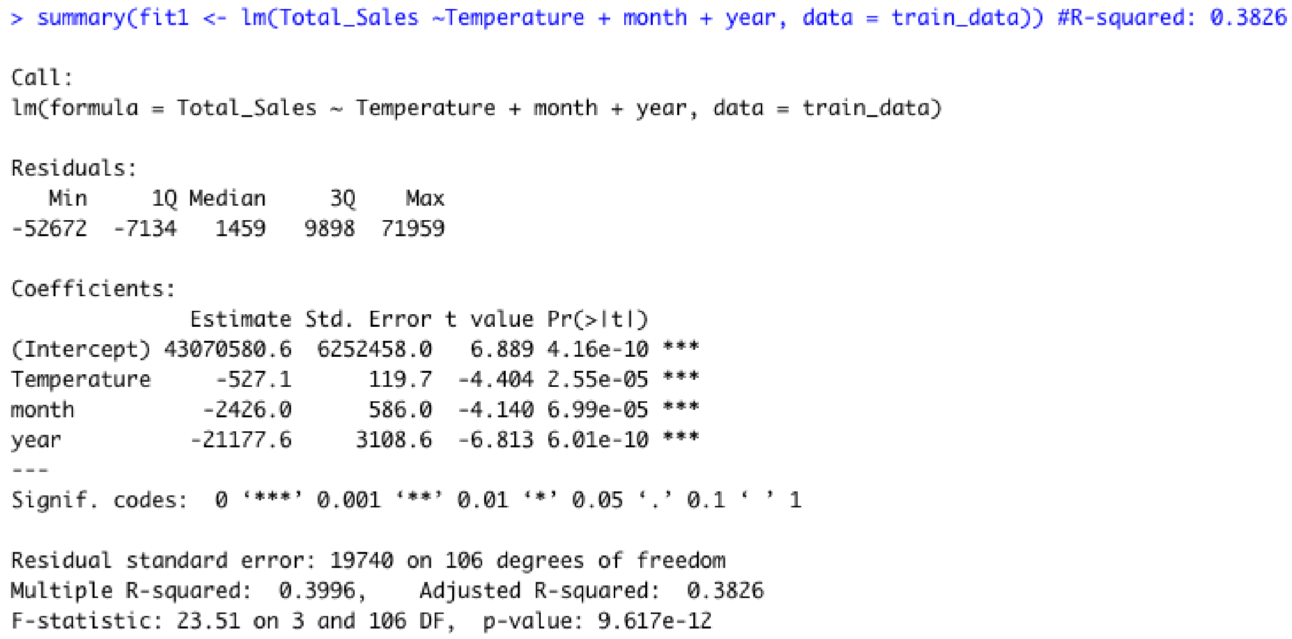


```
> #Fitting the model
> fit <- Arima(train$Total_Sales[train.id],c(1,0,1),xreg=cbind(train$Temperature,train$MarkDown2,train$MarkDown
3,train$month)[train.id,])
> fit
Series: train$Total_Sales[train.id]
Regression with ARIMA(1,0,1) errors

Coefficients:
         ar1      ma1  intercept      xreg1    xreg2   xreg3     xreg4
      0.5862  -0.3809  2251282.5  -8047.441  -5.7937  5.2657  45763.04
s.e.  0.3214   0.3520   115095.4   1983.604   2.6501  2.4963  12318.91

sigma^2 estimated as 6.072e+10:  log likelihood=-1518.13
AIC=3052.25   AICc=3053.68   BIC=3073.86
>
```

The forecast result :



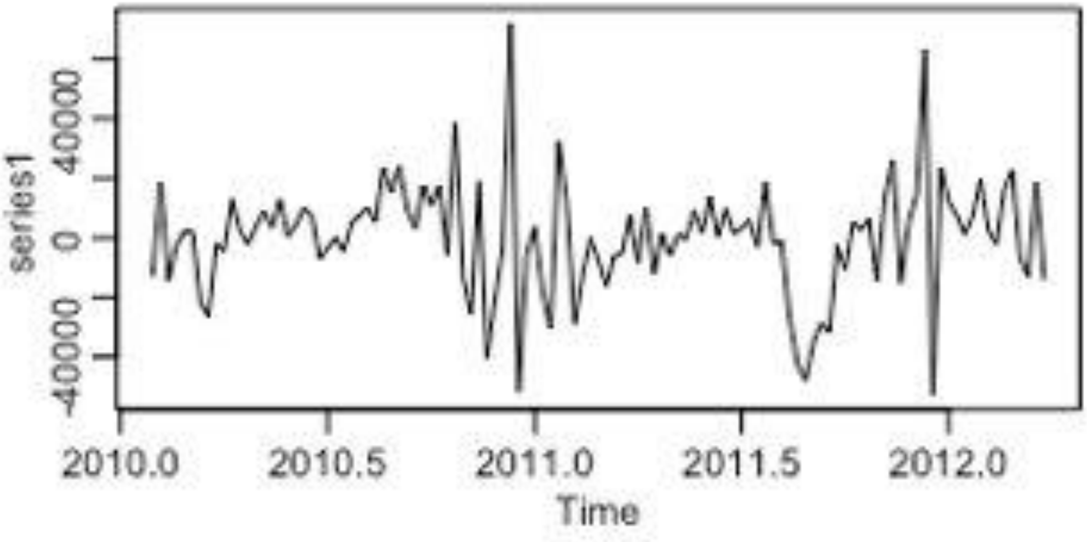
Store 30 :



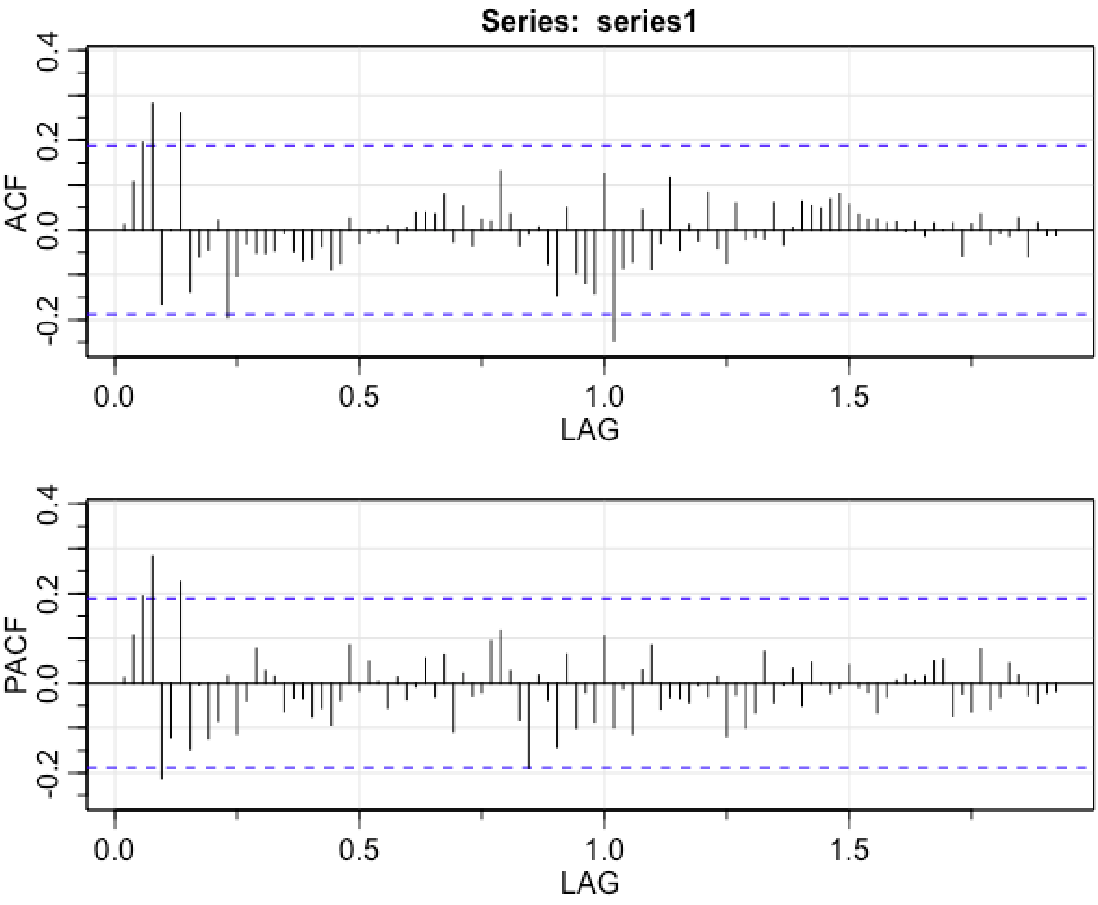
```
> summary(fit1 <- lm(Total_Sales ~Temperature + month + year, data = train_data)) #R-squared: 0.3826

Call:
lm(formula = Total_Sales ~ Temperature + month + year, data = train_data)

Residuals:
   Min     1Q Median     3Q    Max 
-52672  -7134   1459   9898  71959 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 43070580.6  6252458.0   6.889 4.16e-10 ***
Temperature     -527.1      119.7  -4.404 2.55e-05 ***
month          -2426.0      586.0  -4.140 6.99e-05 ***
year          -21177.6     3108.6  -6.813 6.01e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19740 on 106 degrees of freedom
Multiple R-squared:  0.3996,	Adjusted R-squared:  0.3826 
F-statistic: 23.51 on 3 and 106 DF,  p-value: 9.617e-12
```

As we see that the p-value for all the coefficients are smaller, this model was used for extracting the residuals.

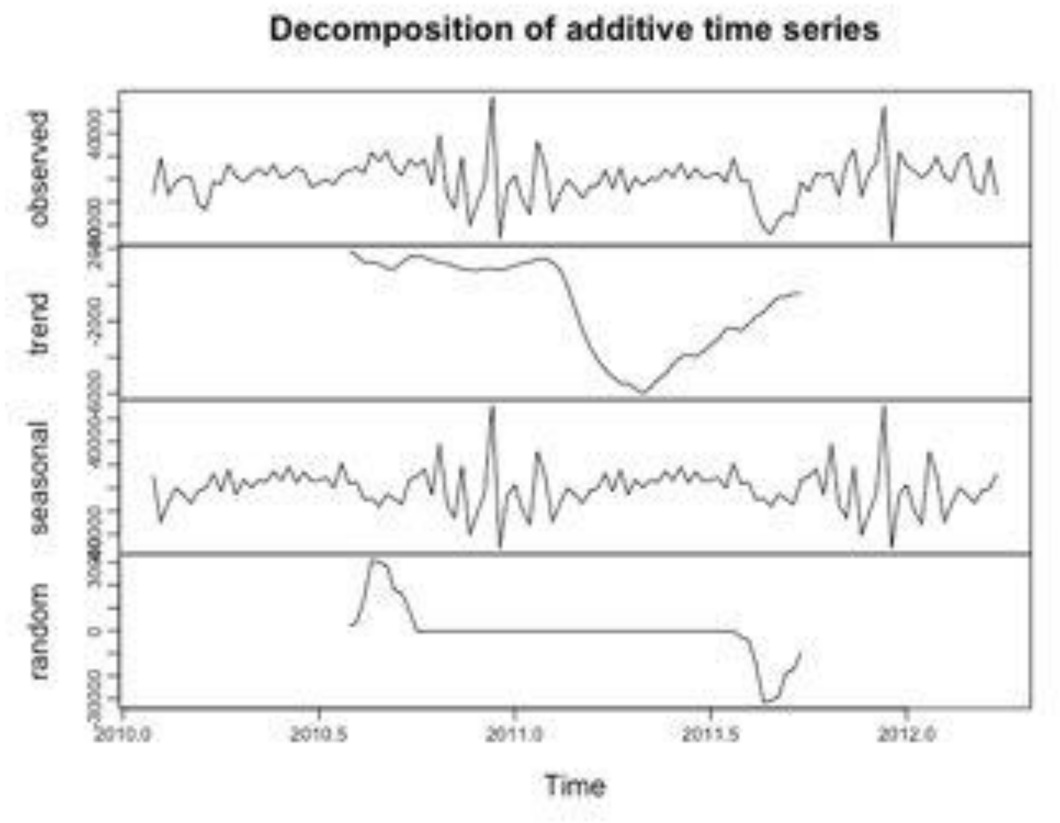
Following was the residual plot for the Store after running the regression model :



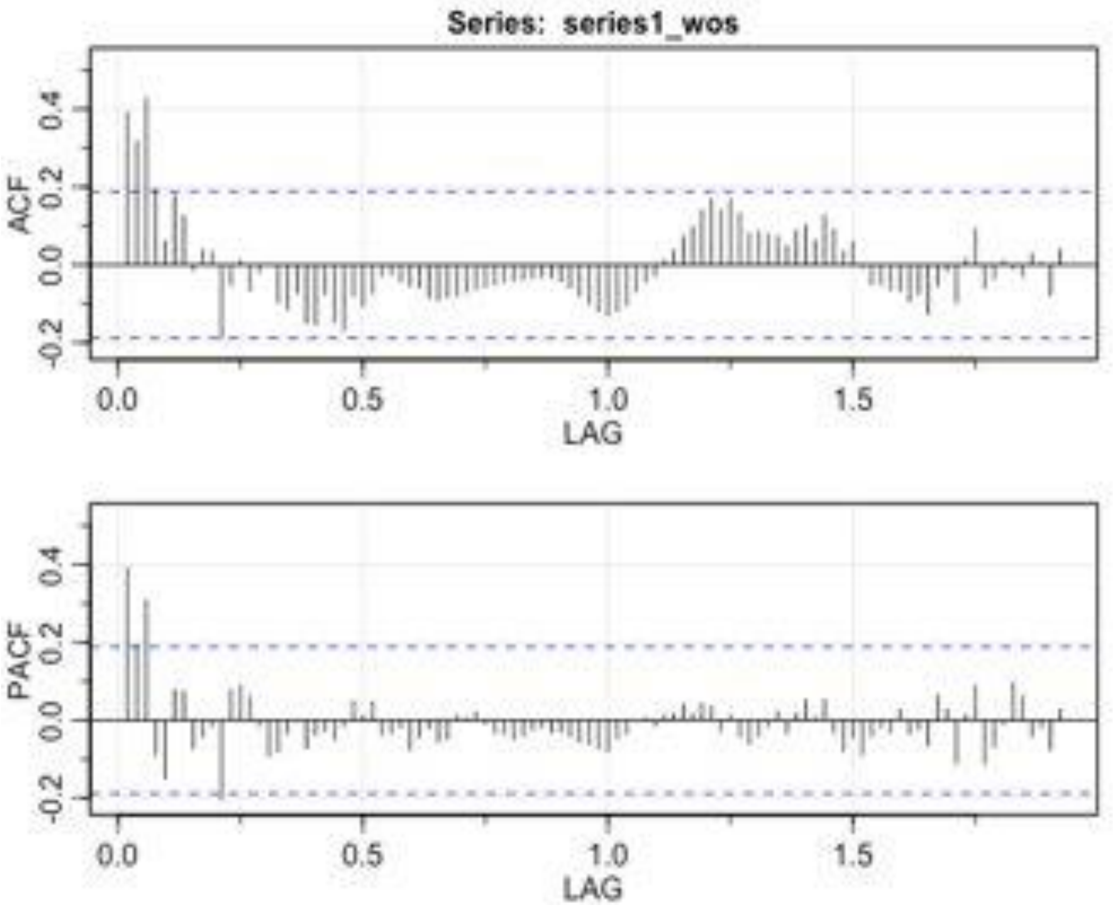
The acf plot for the residual looks like the following :



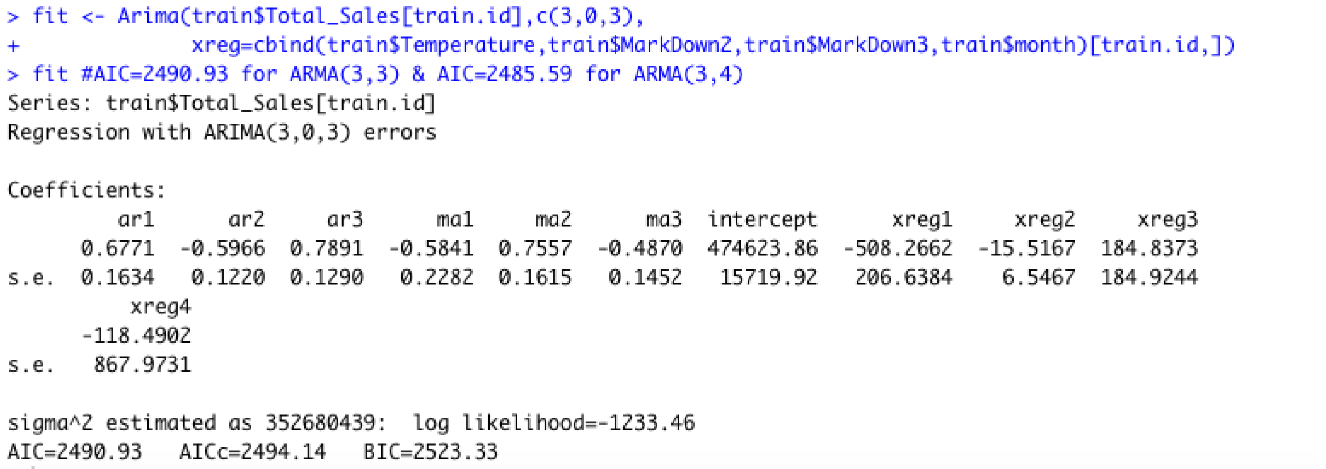
In the acf plot, there is a seasonality component at around 52 weeks, but as the data was limited to 143 weeks, it was not possible to use this seasonality component. So, we decided to remove the seasonality component by running the decomposition and subtracting the component.



After removing seasonality, below were the ACF plots :

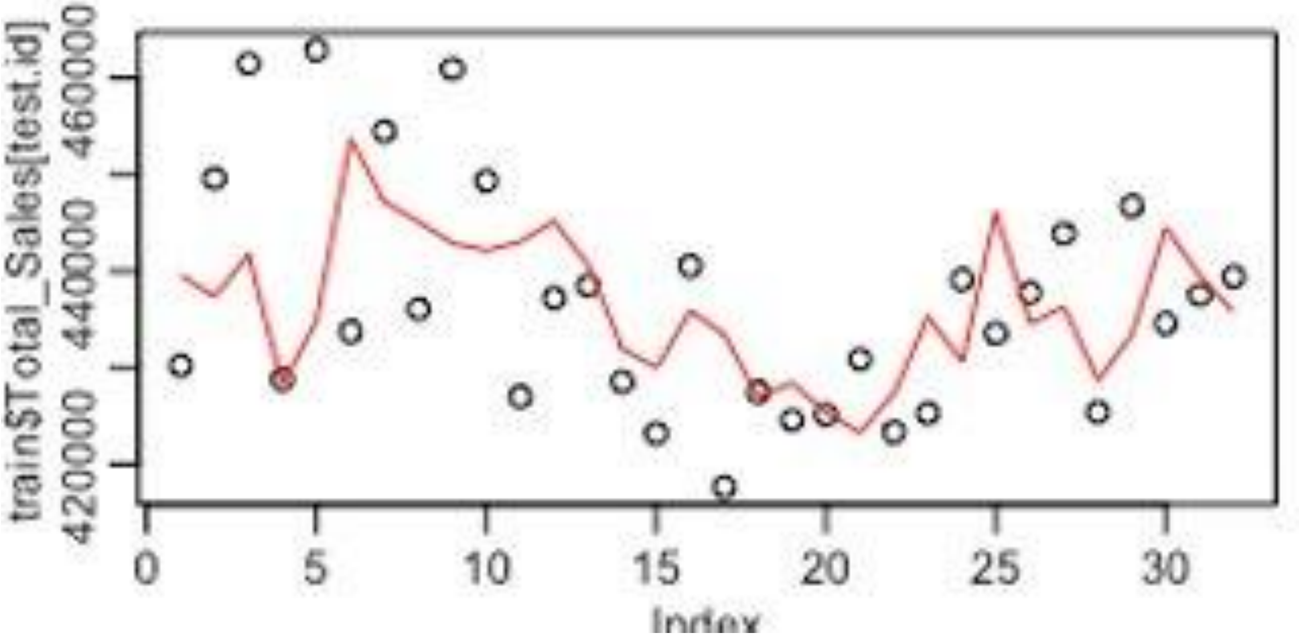


After examining the graph and trying different models, we selected AR3 MA3 process and ran the ARIMA function :



```
> fit <- Arima(train$Total_Sales[train.id],c(3,0,3),
+              xreg=cbind(train$Temperature,train$MarkDown2,train$MarkDown3,train$month)[train.id,])
> fit #AIC=2490.93 for ARMA(3,3) & AIC=2485.59 for ARMA(3,4)
Series: train$Total_Sales[train.id]
Regression with ARIMA(3,0,3) errors

Coefficients:
         ar1      ar2     ar3      ma1     ma2      ma3  intercept      xreg1     xreg2     xreg3
      0.6771  -0.5966  0.7891  -0.5841  0.7557  -0.4870  474623.86  -508.2662  -15.5167  184.8373
s.e.  0.1634   0.1220  0.1290   0.2282  0.1615   0.1452   15719.92   206.6384    6.5467  184.9244
         xreg4
      -118.4902
s.e.   867.9731

sigma^2 estimated as 352680439:  log likelihood=-1233.46
AIC=2490.93   AICc=2494.14   BIC=2523.33
```

The forecast result :



Comparison across 3 stores :

	Store3	Store 20	Store 30
MAPE	0.4440297	0.5241984	0.05970627

MAPE- wise Store 30 tends to perform the best. We can say that ARIMA with residuals is giving us a fair result.

BSTS

BSTS stands for Bayesian structural time series. The advantages of BSTS models are that they can overcome the limitations faced by conventional methods of SARIMA models like handling uncertainty, less data and considering seasonality part in forecasting.

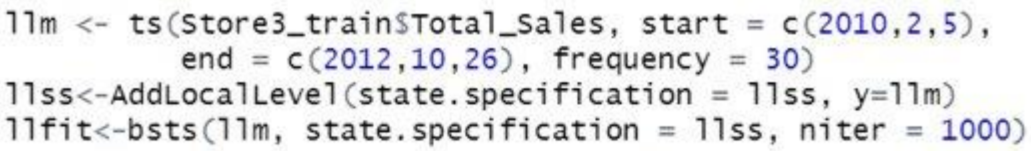
Bayesian models are transparent than ARIMA models as they do not rely on differencing, lags and moving averages. Instead they combine information of prior and estimate parameters from posterior probability distributions. This way it handles the uncertainty associated with forecasting.

We tried fitting bsts models for our 3 stores as selected above.

STORE Type B - 3

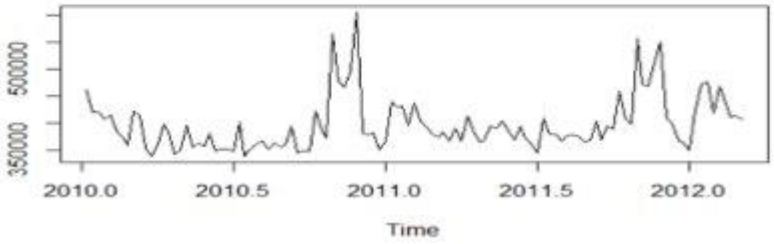
Local Level model

We first fit just the local level model and forecast the time series

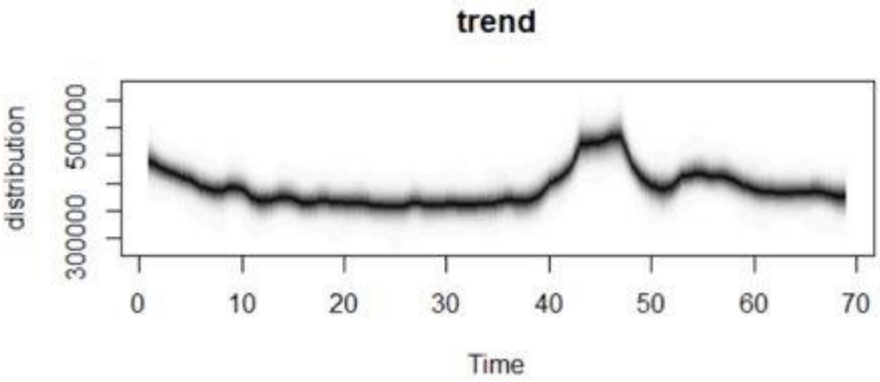


```
llm <- ts(Store3_train$Total_Sales, start = c(2010,2,5),
          end = c(2012,10,26), frequency = 30)
llss<-AddLocalLevel(state.specification = llss, y=llm)
llfit<-bsts(llm, state.specification = llss, niter = 1000)
```

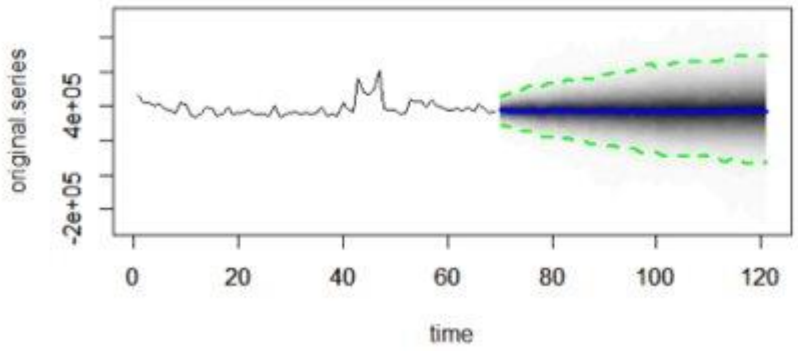
The model looks like below:



When the components are plotted, we can see only trend is captured

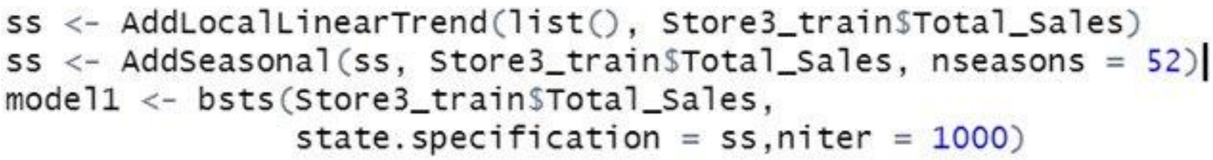


Prediction also didn’t seem to do a very good job. The forecasting for one year just shows prediction around mean



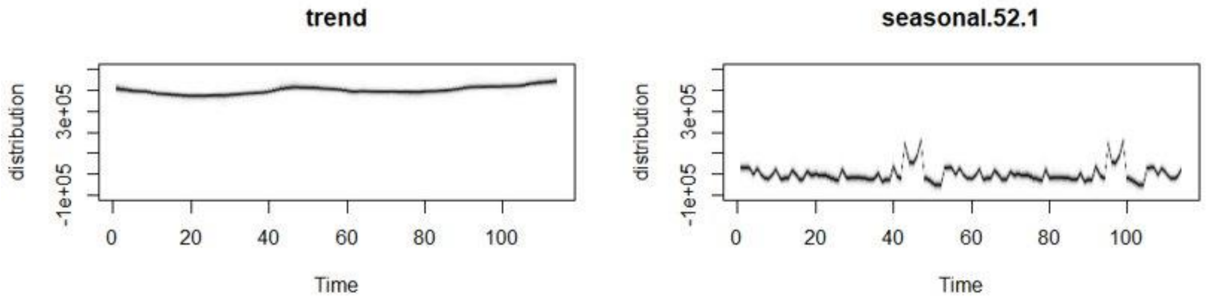
We now fit models adding trend, seasonality and regression components

Model 1: Adding linear trend and seasonality

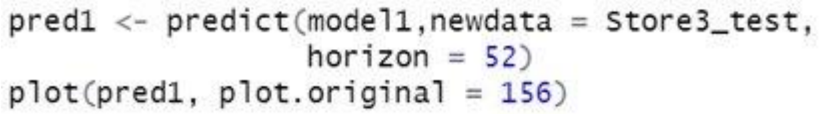


```
ss <- AddLocalLinearTrend(list(), Store3_train$Total_Sales)
ss <- AddSeasonal(ss, Store3_train$Total_Sales, nseasons = 52)
model1 <- bsts(Store3_train$Total_Sales,
               state.specification = ss,niter = 1000)
```

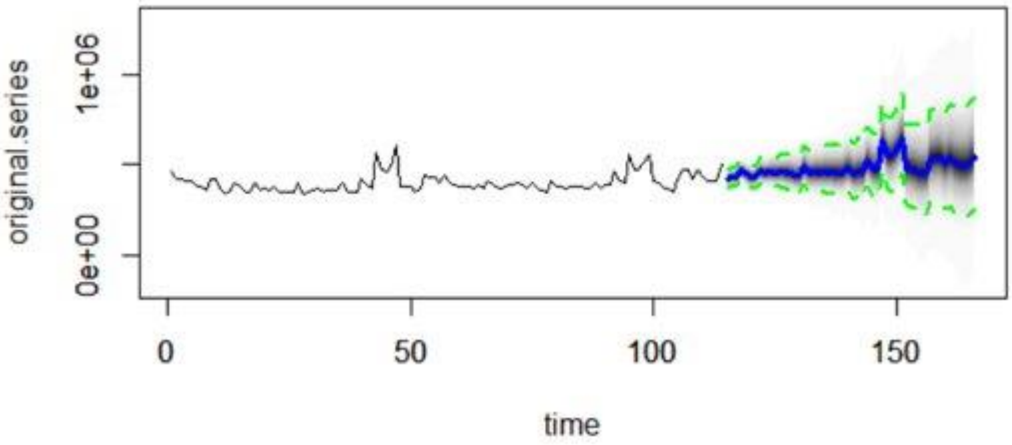
Model 1 – components



Model 1 – prediction



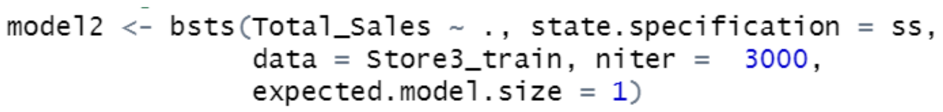
```
pred1 <- predict(model1,newdata = Store3_test,
                 horizon = 52)
plot(pred1, plot.original = 156)
```



We can see that the model is now able to give prediction taking trend and seasonality into consideration

We now add regression component as well into the model.

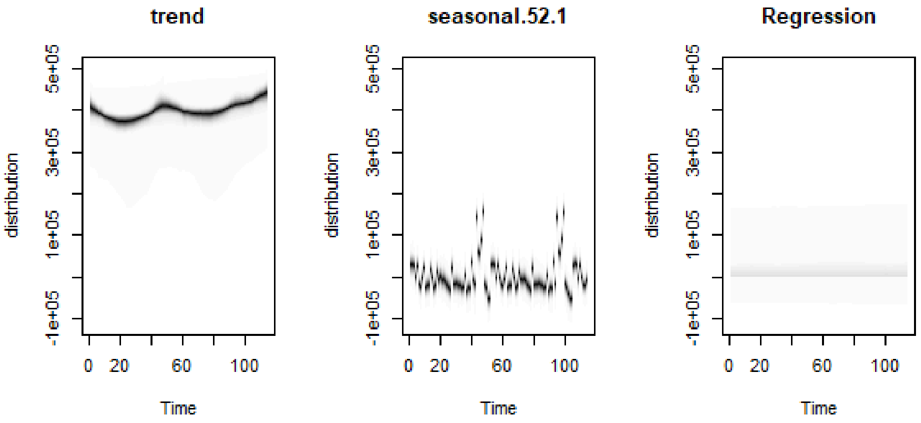
Model 2: Trend, Seasonality and Regression with expected size=1



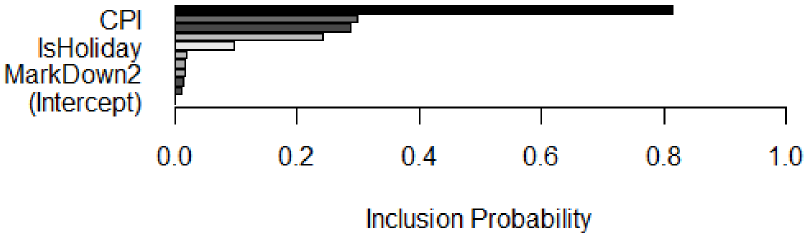
```
model2 <- bsts(Total_Sales ~ ., state.specification = ss,
               data = Store3_train, niter =  3000,
               expected.model.size = 1)
```

Expected model size =1 sets the spike and slab prior to have one spike. In other words, we are expecting one variable to heavily influence our weekly sales

Model 2 – components

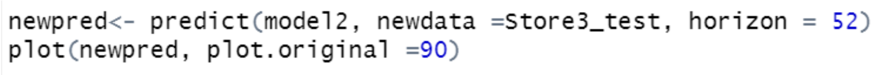


One of the advantages of BSTS is it also gives us important features based on their inclusion probability. From below plot, CPI seems to heavily influence our weekly sales

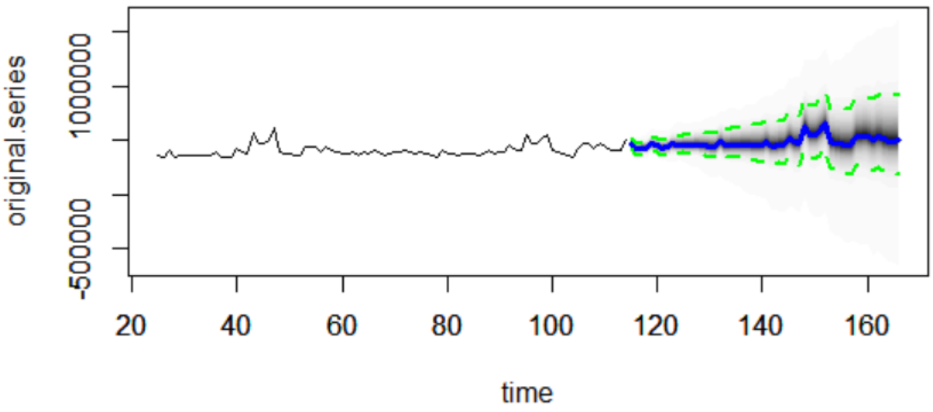


Model 2 – prediction

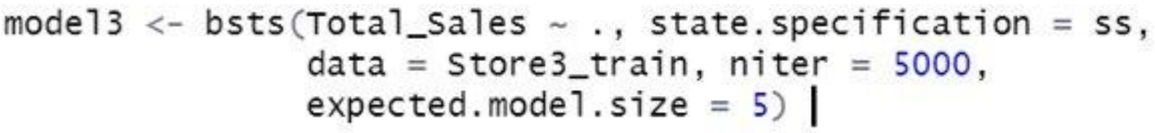
We predicted for one year (52 weeks) using information of last 90 weeks



```
newpred<- predict(model2, newdata =Store3_test, horizon = 52)
plot(newpred, plot.original =90)
```



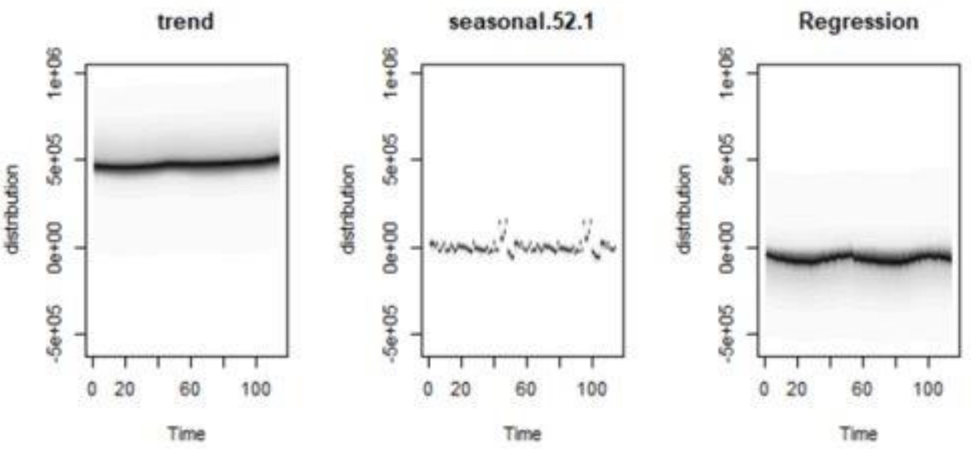
Model 3: Trend, Seasonality and Regression with expected size=5



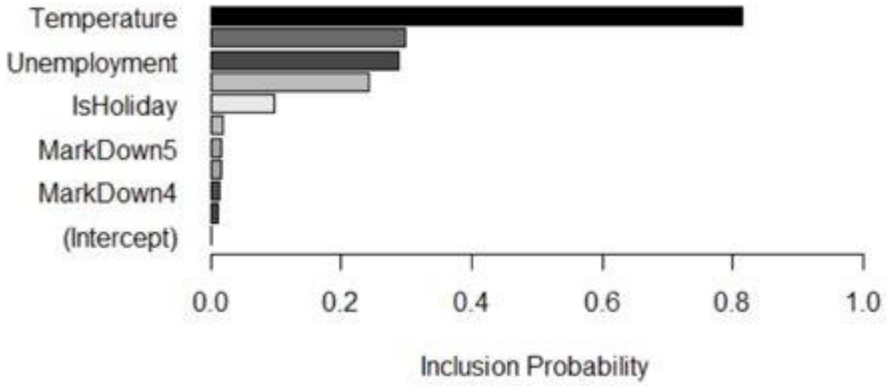
```
model3 <- bsts(Total_Sales ~ ., state.specification = ss,
               data = Store3_train, niter = 5000,
               expected.model.size = 5)
```

Expected model size =1 sets the spike and slab prior to have 5 spikes. In other words, we are expecting 5 variables to mostly influence our weekly sales

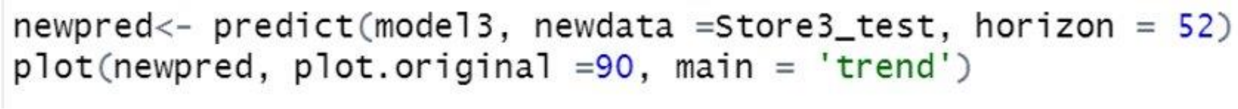
Model 3 – Components



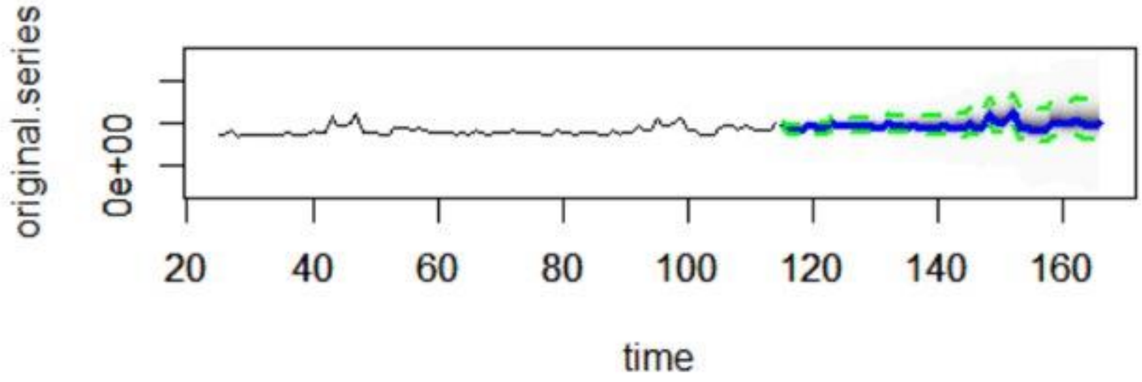
Important features



Model 3 - Prediction

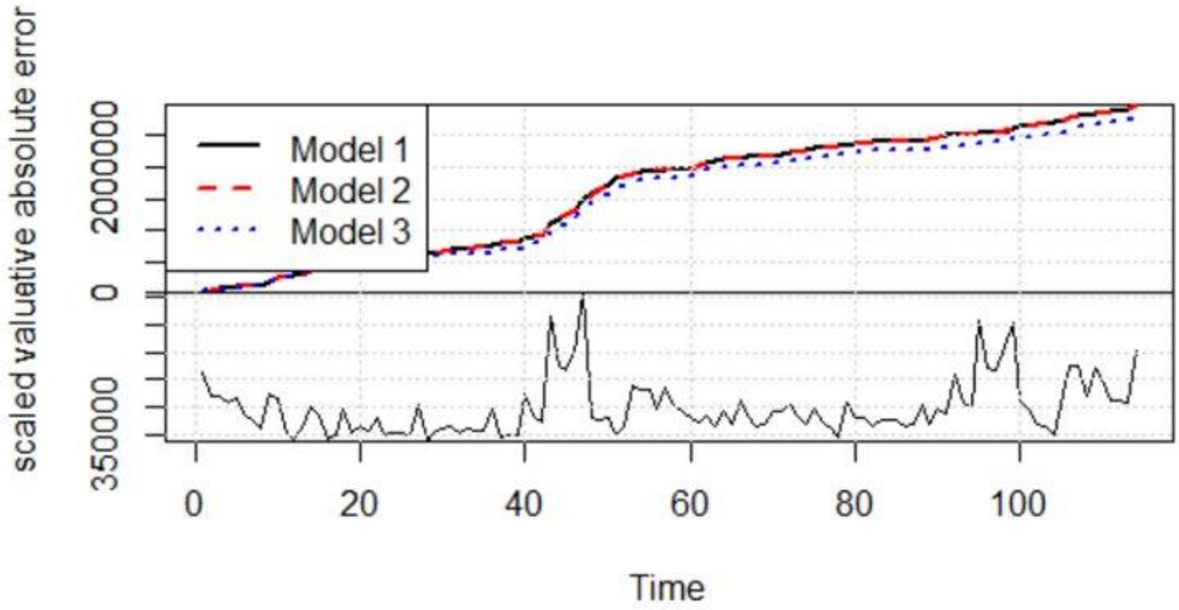


```
newpred<- predict(model3, newdata =Store3_test, horizon = 52)
plot(newpred, plot.original =90, main = 'trend')
```



We can see the model 3 gave prediction well with narrow confidence interval.

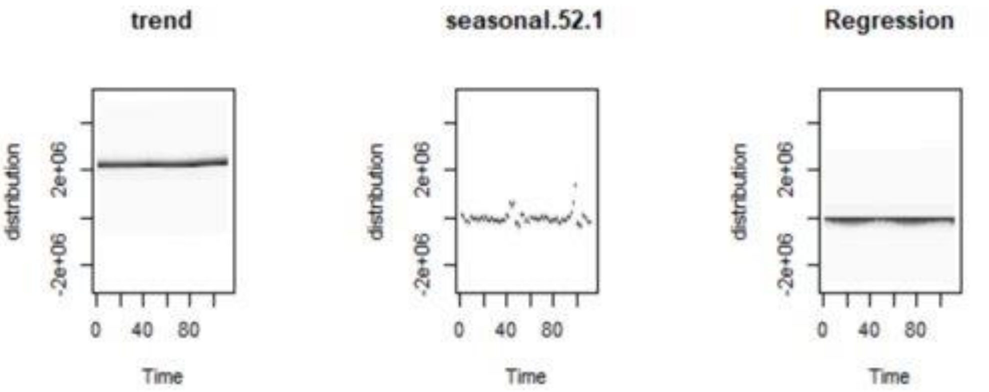
Comparison of models:

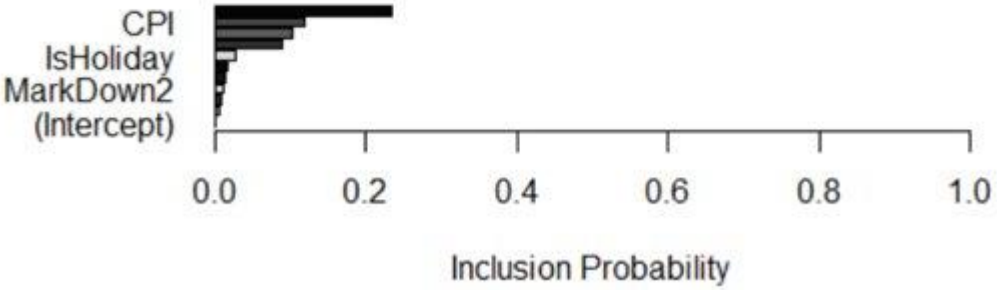


Similar models are run for other store types as well

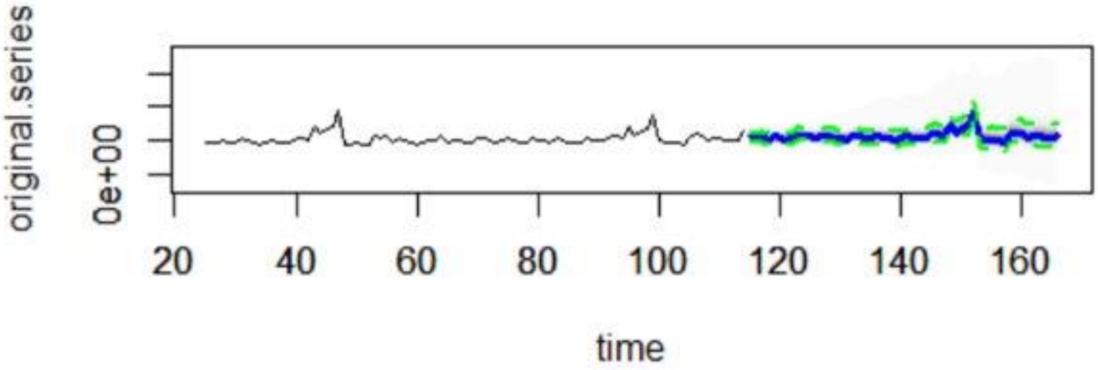
Store Type A – 20

Components



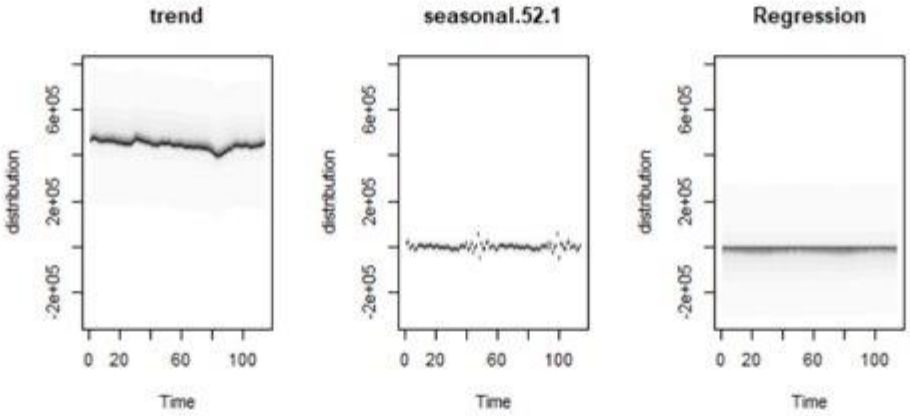


Prediction

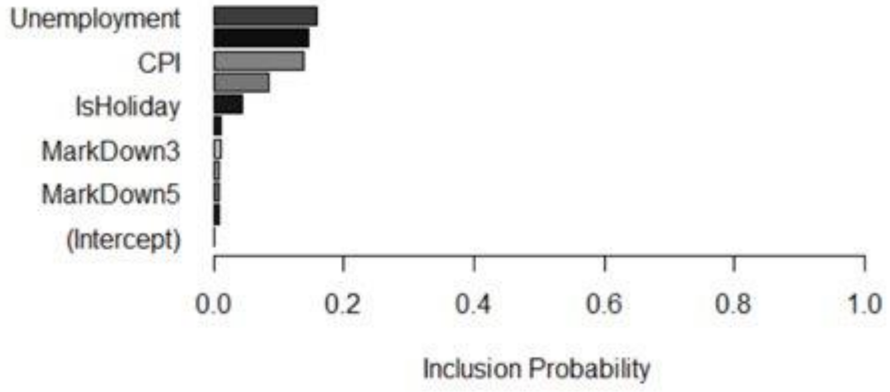


Store Type C – 30

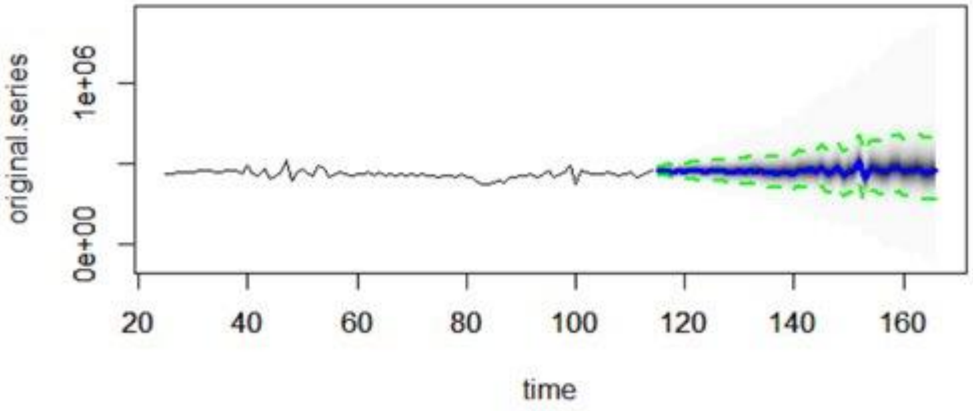
Components



Important features



Prediction



Comparison across 3 stores :

For BSTS:

	Store3	Store 20	Store 30
MAPE	14.10561	16.66228	5.047219

MAPE- wise Store 30 tends to perform the best.

CONCLUSION

In Conclusion, our SARIMA model gave a fairly good result with a MAPE of under 4 %.. However, while trying SARIMA with additional variables we did not get the seasonality right as the period was too lest for repeat seasonality. Hence, we decided to remove the seasonality component by using the decomposition method.

Running the ARIMA model on all the three stores gave a MAPE of under 0.4 % however we proceeded to BSTS which results in terms of giving predictions taking seasonality into consideration with a MAPE of under 16 %. Because the data doesn't span over more years, the seasonality that was taken care by this method is proving to give not so good results like the other two methods.

NEXT STEPS

We would like to continue working on our models and follow further steps to make them better. Here are few next steps:

1. We could fine tune our models by playing around with the model parameters
2. Test one store each from the same store type category we took already and fit the model to test accuracy
3. Extend the analysis for all stores
4. Develop department wise forecast models and look at future performance of various departments

TEAM CONTRIBUTION :

1. The team sat together to brainstorm on the exploratory data analysis and how the data could be dissectioned to derive most insights. We then divided the project into the following sections and picked up each part.
    a. SARIMA without variables
    b. SARIMA with variables
    c. ARIMA with variables
    d. BSTS

We shared the results and the problem in each section and tried to reach to a solution. The whole team contributed equally to the success of this project.