Predicting Sales: Corporacion Favorita

Team Member: Sanket Badhe, Parth Shah, Isha Raju

Introduction

Driving sales is the most essential part to brick-and-mortar retailers. Successful retailers not only rely on great customer service but also on key strategic decisions, performance metrics and sales history. Data-driven insights can help make better as well as reliable business decisions faster.

Corporacion Favorita is a large Ecuadorian-based grocery retailer, operating hundreds of supermarkets with over 200,000 different products on their shelves. Being able to forecast sales based on factors such as history of items sold and sales prices can help the retail giant accurately assess the number of sales for a particular store. Our goal is to identify trends and seasonality to fit a time-series model.

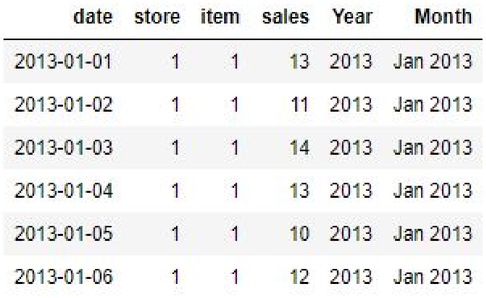
Biggest challenge in this competition is to predicts sales of each item in each store separately. Sales of most of the products are independent of each other. So we have to build a model for each item separately. We use stack ensembling of time series linear model and ARIMA to overcome this and generalized our time series analysis technique.

Dataset

The dataset is publicly available on Kaggle, provided by the retail chain, Corporacion Favorita, itself. The response variable is the number of *unit sales*, by store and item. The predictor variables are *date* and *store number*. Two new features *month* and *year* are added to the dataset for detailed analysis. The data includes information of 10 stores and 50 different store items. All data accounted for is in a single .csv file:

- **Train.csv:** The primary dataset including sales data by store, item and date from Dec 31st 2012 to Dec30th 2017.
- **Test.csv:** The dataset used for testing the model includes dates from Dec 31st 2017 to March 30th 2018.

R programming has been used for exploratory analysis and modeling.



date	store	item	sales	Year	Month
2013-01-01	1	1	13	2013	Jan 2013
2013-01-02	1	1	11	2013	Jan 2013
2013-01-03	1	1	14	2013	Jan 2013
2013-01-04	1	1	13	2013	Jan 2013
2013-01-05	1	1	10	2013	Jan 2013
2013-01-06	1	1	12	2013	Jan 2013

Exploratory Data Analysis

The training set has 913000 observations and four variables: *date, store, item, sales.* First, it was checked for missing values and extreme outliers. We also visualized the data to take note of the distribution of unit sales and its relationship on a daily, monthly and yearly basis.

The unit sales histogram shows a high peak above 90,000 and then it declines gracefully representing a right-skewed distribution **[1]**. Next, the unit sales growth was plotted with respect to date and an increasing trend was observed from 2013-2017**.** The rate of change in sales price were also accounted on a daily basis, and they remained constant throughout the four years **[2]**. Similarly, the same increasing trend in sales growth was observed on a monthly and yearly basis as well. However, the rate of change in sales price showed fluctuations indicating the presence of seasonality **[3]**. On a yearly basis, the rate change increased substantially from year 2013 to 2014, with a dip in 2015. Then that same pattern was observed from 2015-17 **[4]**. The highest rate in change was in 2014. Following this pattern, our assumption is that the rate change in sales price will increase in 2018.

The dataset was also checked for correlations between sales and date. The highest number of sales were observed in the weekends. While May-Aug were the months showing the most sales activity **[5].**

Methodologies:

We have a total of 10 stores in our data and in each store we have a 50 items. Our objective is to predict the sales of each item in each individual store. So there will be total 500 unique combination of item and store in the data. we have to build the separate time series model for each combination for accurate prediction. So our analysis consists of mainly three parts:

- First, We started our analysis by first focusing one particular item in one of the stores. We build time series model for that particular unique combination.
- Second, Generalize or automize similar techniques to rest of the combinations(store-item) in the data.
- Third, Validation of our automated technique.

First Part

For the first part of our analysis, we randomly selected the single combination of store and item. Time series can be thought as additive model of trend component, seasonal component and random component. Time series decomposition help us to find out deterministic and non deterministic components. Hence, we decomposed our time series into these three components and tried to analyse it. Next we check the stationarity of the time series using Autocorrelation function and Augmented dickey fuller test.

For modelling part we first fit tslm(time series linear model) in built function in the forecast library. This regression based methods to fit time series data including trend and seasonality. We provide seasonality as an input to this function. After fitting the tslm model we analysed its residual. We examined residual plot to ensure the randomness of residual with no trends. We also checked normality of the residual with histogram and qq plot. We also applied shapiro-wilk test of normality on residual. We also checked for ACF of the residual if there is a lack of residual our forecast is good.

After fitting of the regression model we got residuals. But many times these residuals have time series trend. To overcome this we fitted ARIMA model on the residual of Regression Forecast. ARIMA model improves the error performed by Regression time series model. We also analysed ARIMA residual. We examined residual plot to ensure the randomness of residual with no trends. We also checked normality of the residual with histogram and qq plot. We also applied shapiro-wilk test of normality on residual. We also applied Box-Ljung test to test the whiteness of residual.

Final Forecast = Regression Forecast + ARIMA Forecast

Second Part

In the first part we created a time series model for just 1 unique combination of store and item. In the 2nd part, our main objective is to generalize this technique to all the combination of store and item. We created nested loop (i.e. two for loop), one for item and another for store. In each iteration of the loop we fitted Regression model and ARIMA model sequentially. For fitting linear model we don't have to pass any hyperparameter but for ARIMA model we need to pass hyperparameter i.e. order (p,d,q) while running it. While looking for the solution of this issue we stumble upon the auto.arima function in forecast package. This function automatically search for the order of ARIMA model. Finally we predicted our Final Forecast in each iteration(i.e each unique combination of item, store) as summation of Regression Forecast and ARIMA forecast.

Third Part

We validated our model on test data using built model and submitted our answer on Kaggle. We also attached a copy of our answer in CSV format with other files. We also calculated MAPE(mean absolute percentage error of our model).

Results:

First Part:

We randomly selected the 4th item of the 5th store for the first part of the analysis. We plotted the sales of selected item with time using autoplot **[6]**. Then we created a decomposed plot of trend, seasonality and random of the selected item **[7]**. In the trend plot we can see the increasing trend and In the seasonality plot we can see the monthly seasonality. In the next part we plotted the ACF of sales of this functions **[8]**. From the ACF plot we can conclude that the

Autocorrelation of the sales are significant for all the lag. For checking stationary, we ran Augmented dickey fuller test and we got p value as 0.01. Due to small p value of ADF test we reject the null hypothesis i.e. ADF test suggest that sales data are stationary time series for this particular item. Due to contrasting result of ADF test and ACF plot we can conclude that our time series is trend stationary.

Next, we fitted a time series linear model to capture trend and seasonality. From summary **[9]** of the table we can see that the p value for trend is really significant and p value for variable related to seasonality is really high. Hence, our model captured the trend in the model. Then we focused on residual analysis of our linear model. In the ACF plot of the residual **[10]** all the autocorrelation for all lags was smaller than significant value. Hence, the lack of autocorrelation shows that our linear model is a good forecast predictor. Then we examined normality with the help of QQPlot and Histogram of Residual plot **[11]**.Both of these plots supports the normality of the residual. We also ran Shapiro-Wilk test for checking normality. Because of large p value of Shapiro-Wilk test we cannot reject the null hypothesis that is residual is normally distributed. Lastly we visually examined residual plot**[12]**, from the residual plot we concluded that there is no pattern in residuals and residual are random around mean 0.

Going forward, we fitted ARIMA on residual of time series linear model using auto.arima function. Order of our ARIMA was 0, 0, 0 we expected the same result because forecast error from our time series linear model was white noise. i.e. we captured all the deterministic component of our time series. We examined the ACF of residual of our ARIMA model and the only statistically significant correlation was at lag 24**[13]**. Next we ran Box-Ljung test for this model and p value was greater than 0.05. Hence, Box-Ljung test for this model fails to reject the hypothesis of residual whiteness. Hence, we got more indication that the model has captured the dependence in the time series. We also tested QQPlot and Histogram of residual both of them support normality.

Second Part:

We ran generalized time series model as we discussed in the methodology section of this report for every item of every store. Then we picked up item 4 of store 10 randomly and analysed the built model. I applied all the methods similar to part one of our result section. This time linear model was not perfect as we have some values on the ACF plot are greater than the confidence interval. Also, normality assumption was not strictly followed. But after applying ARIMA model on residual plot of the linear model model is significantly improved. ACF plot of the residual all the autocorrelation for all lags was smaller than significant value and Box-Ljung test for this model fails to reject the hypothesis of residual whiteness.

Third Part:

We calculated MAPE(mean absolute percentage error of our model) for all the combinations of store and item.

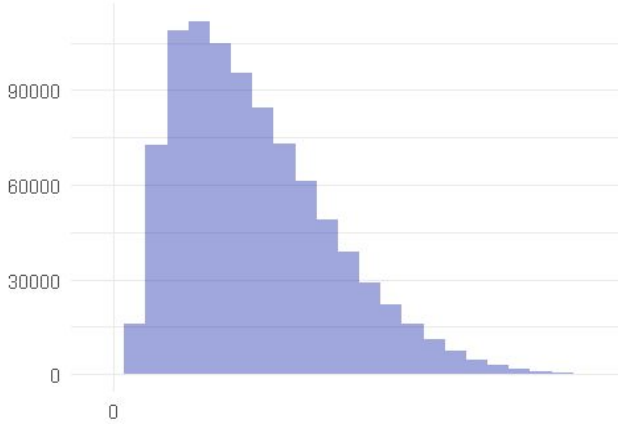
Conclusion:

This study of Corporacion Favorita sales data showed that we can forecast the future sales using time series analysis. This study also showed that sales forecasting with real world data is really complicated and that's why we used stack ensemble method for time series prediction. Also, our model is built on really limited information and there is scope of improvement in modelling results by involving different sources of data. For example: we can include oil prices data since Ecuador is oil dependent country. Lot of the sales are dependent on Holidays and other major events such as FIFA. Also, recent advanced research paper showed improvement in result by utilizing advanced modelling techniques such as Gradient Boosting, LSTM, prophet(link in reference).

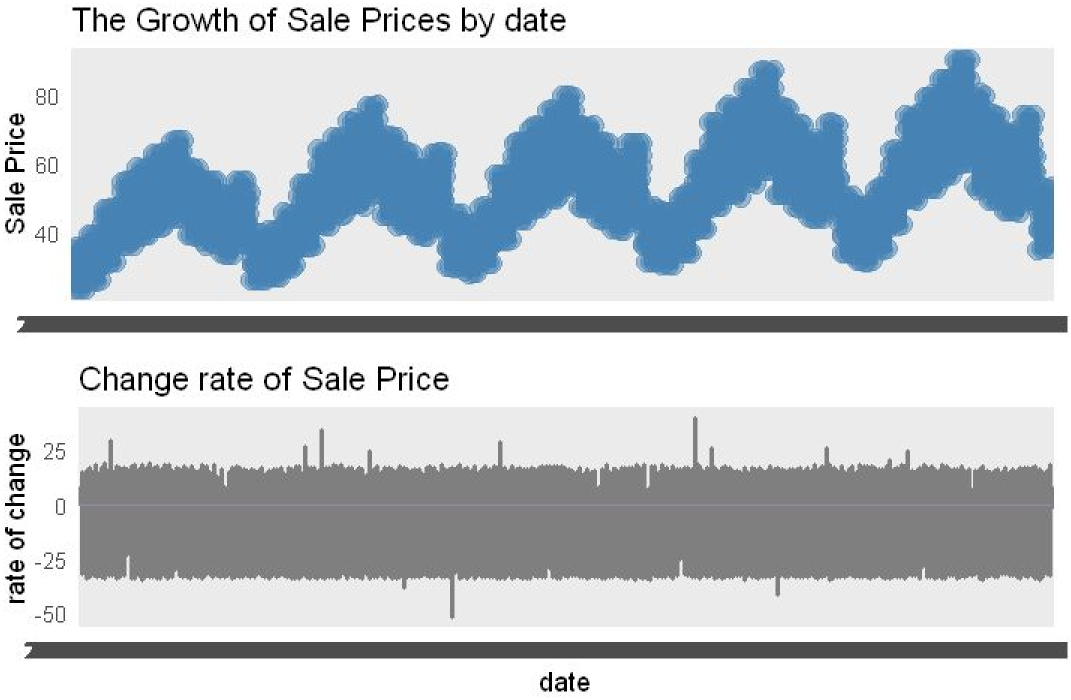
References:

1) https://facebook.github.io/prophet/
2) https://www.kaggle.com/c/favorita-grocery-sales-forecasting

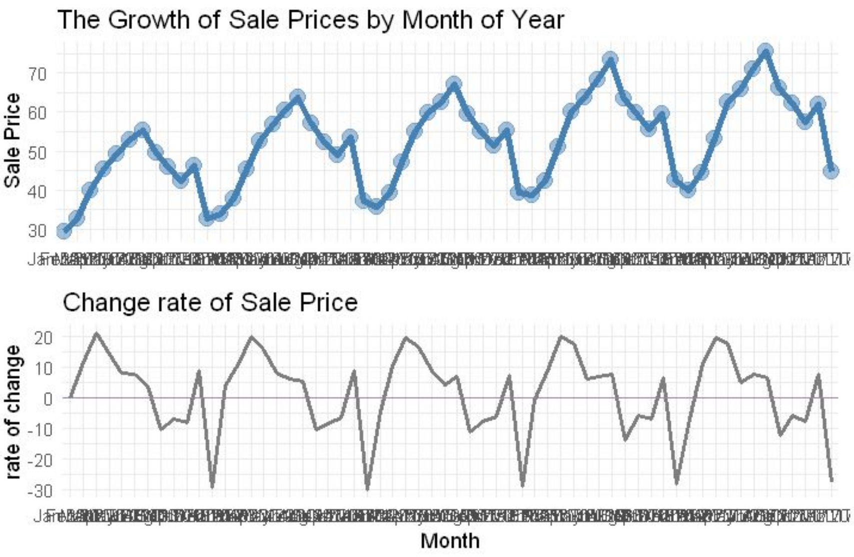
[1]


90000
60000
30000
0
0

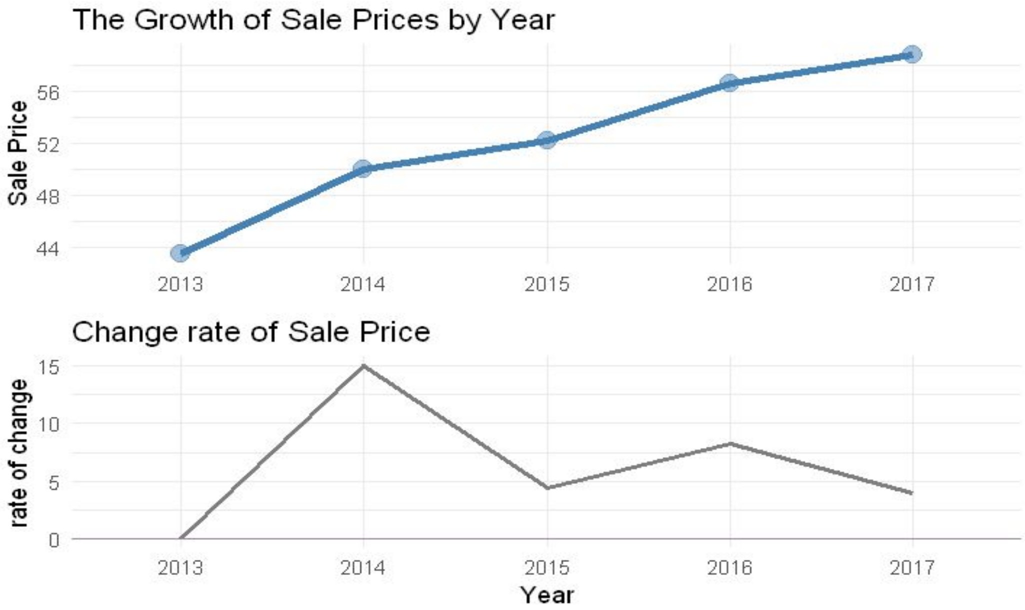
[2]


The Growth of Sale Prices by date
Sale Price
80
60
40
Change rate of Sale Price
rate of change
25
0
-25
-50
date

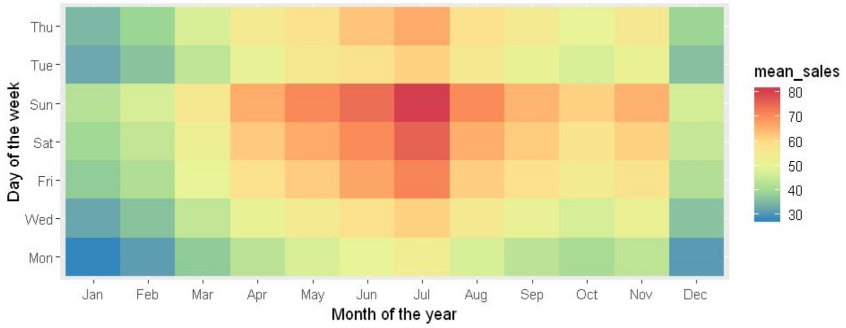
[3]


The Growth of Sale Prices by Month of Year
Sale Price
70
60
50
40
30
Change rate of Sale Price
rate of change
20
10
0
-10
-20
-30
Month

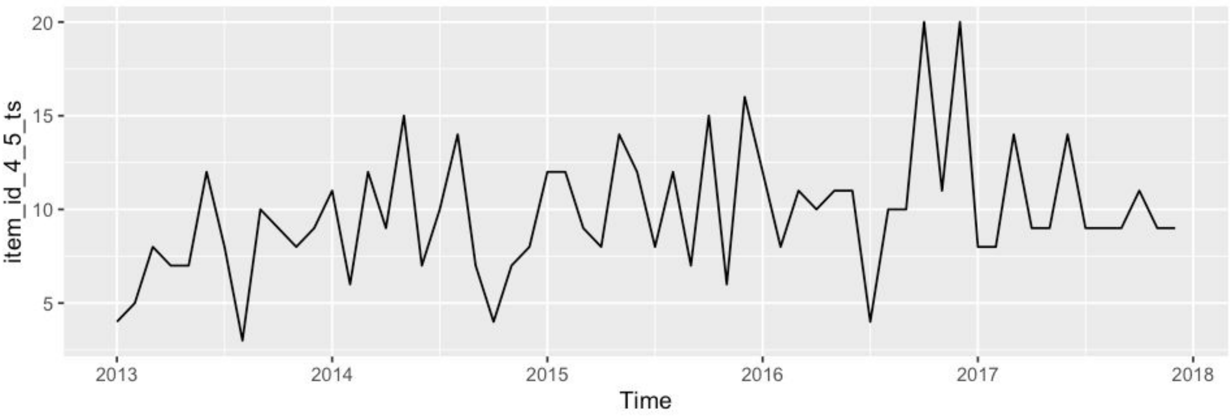
[4]


The Growth of Sale Prices by Year
Sale Price
56
52
48
44
2013
2014
2015
2016
2017
Change rate of Sale Price
rate of change
15
10
5
0
2013
2014
2015
2016
2017
Year

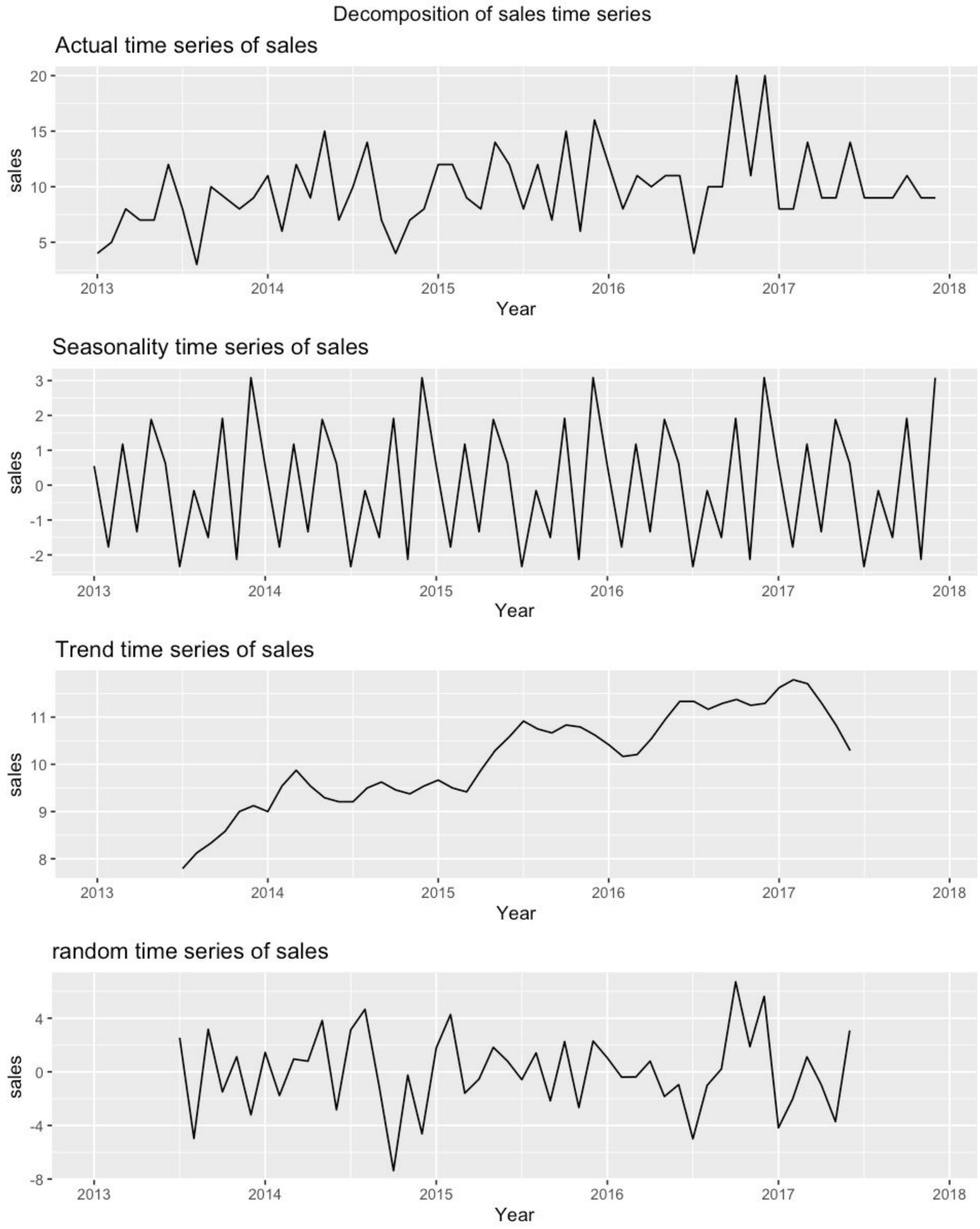
[5]


Thu
Tue
Sun
Sat
Fri
Wed
Mon
Day of the week
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month of the year
mean_sales
80
70
60
50
40
30

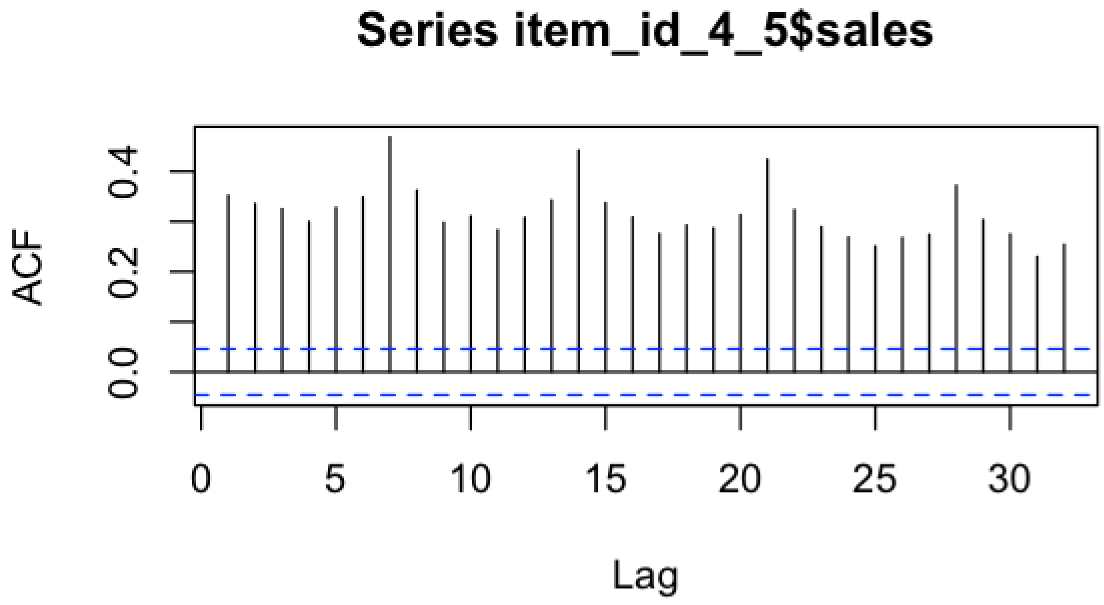
[6]



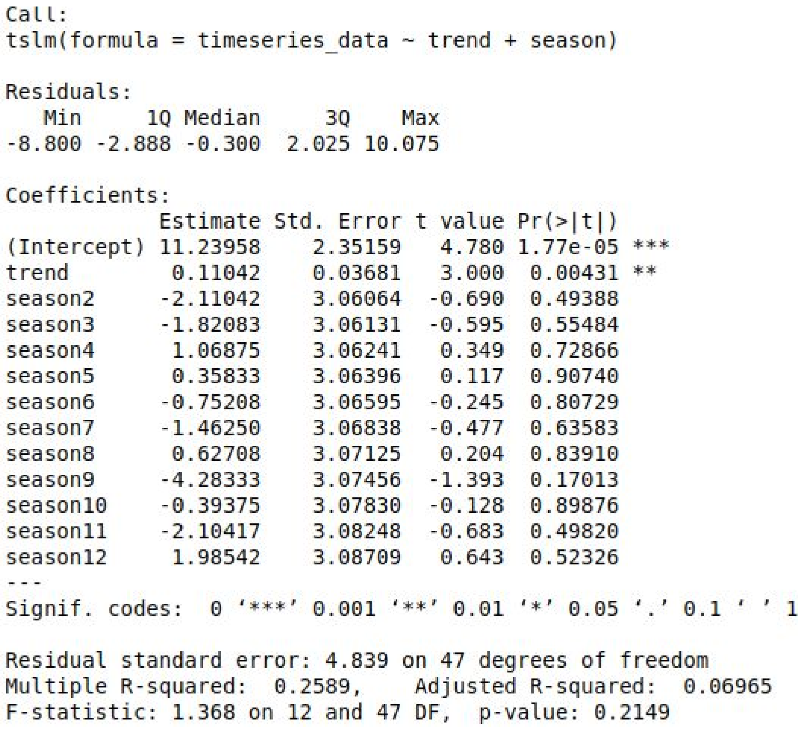
[7]


Decomposition of sales time series
Actual time series of sales
sales
20
15
10
5
2013
2014
2015
2016
2017
2018
Year
Seasonality time series of sales
sales
3
2
1
0
-1
-2
2013
2014
2015
2016
2017
2018
Year
Trend time series of sales
sales
11
10
9
8
2013
2014
2015
2016
2017
2018
Year
random time series of sales
sales
4
0
-4
-8
2013
2014
2015
2016
2017
2018
Year

[8]

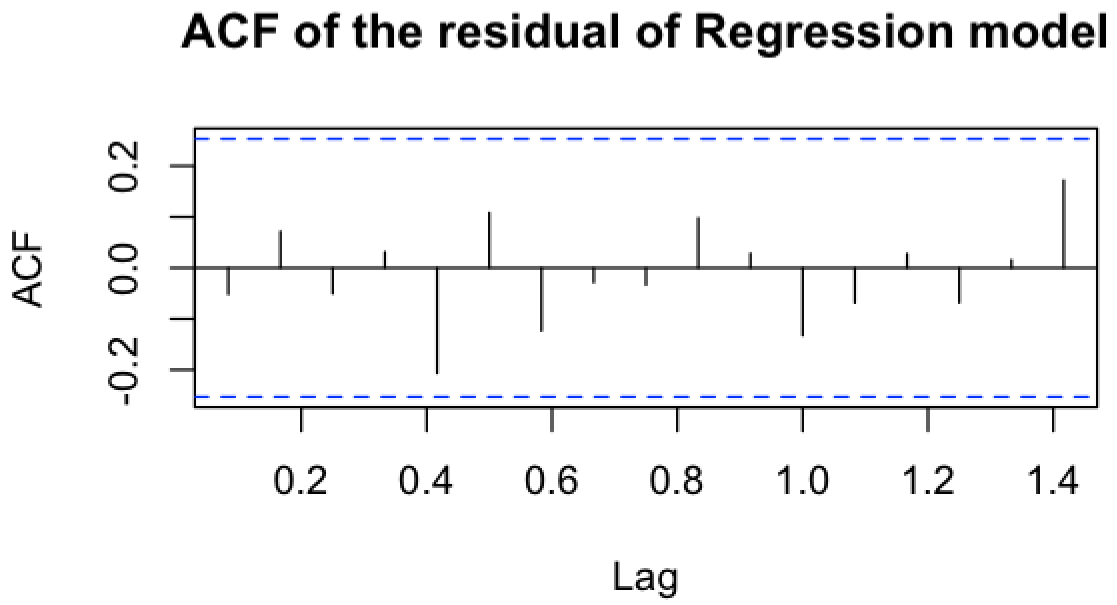


[9]



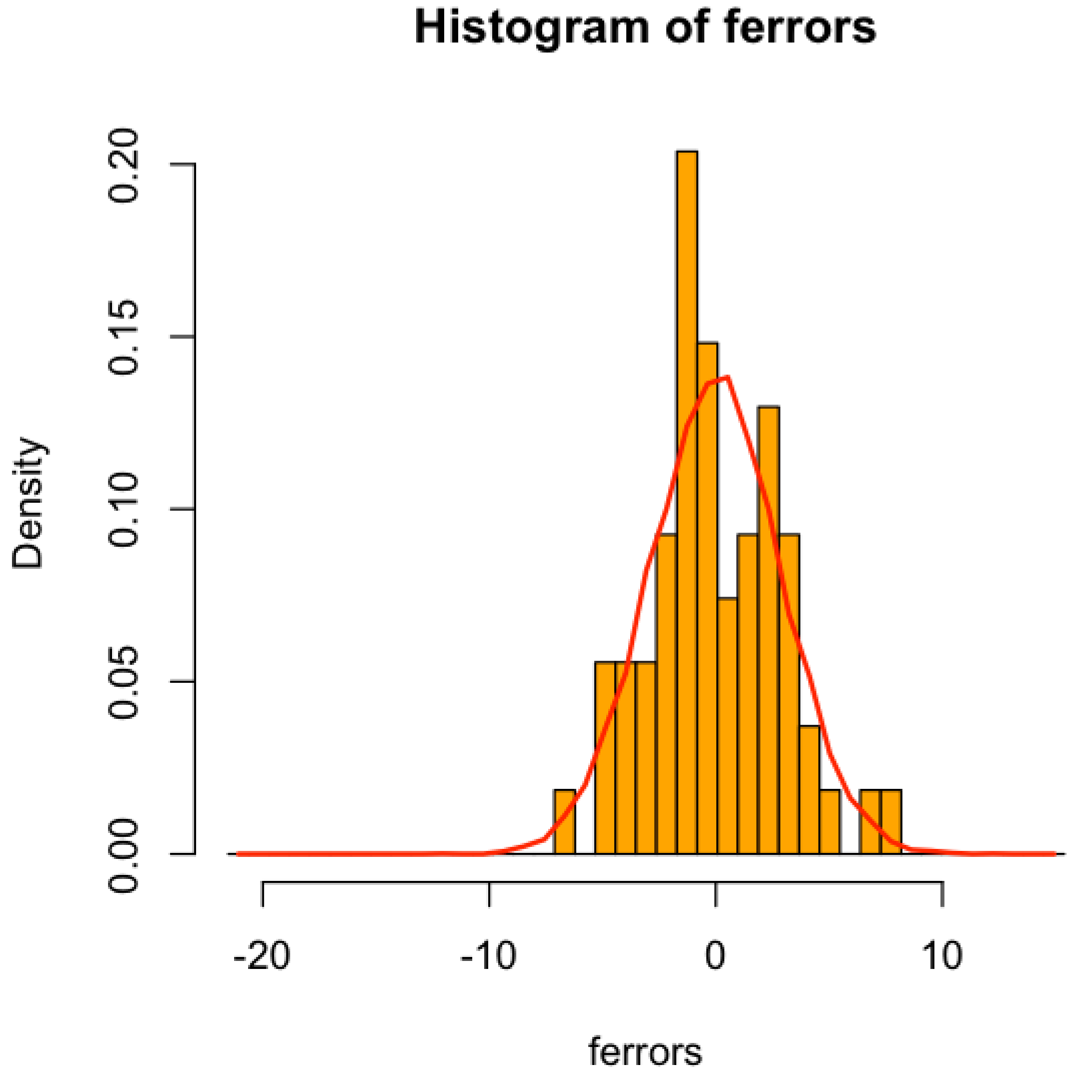
```
Call:
tslm(formula = timeseries_data ~ trend + season)

Residuals:
   Min     1Q Median     3Q    Max 
-8.800 -2.888 -0.300  2.025 10.075 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 11.23958    2.35159   4.780 1.77e-05 ***
trend        0.11042    0.03681   3.000  0.00431 ** 
season2     -2.11042    3.06064  -0.690  0.49388    
season3     -1.82083    3.06131  -0.595  0.55484    
season4      1.06875    3.06241   0.349  0.72866    
season5      0.35833    3.06396   0.117  0.90740    
season6     -0.75208    3.06595  -0.245  0.80729    
season7     -1.46250    3.06838  -0.477  0.63583    
season8      0.62708    3.07125   0.204  0.83910    
season9     -4.28333    3.07456  -1.393  0.17013    
season10    -0.39375    3.07830  -0.128  0.89876    
season11    -2.10417    3.08248  -0.683  0.49820    
season12     1.98542    3.08709   0.643  0.52326    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.839 on 47 degrees of freedom
Multiple R-squared:  0.2589,	Adjusted R-squared:  0.06965 
F-statistic: 1.368 on 12 and 47 DF,  p-value: 0.2149
```

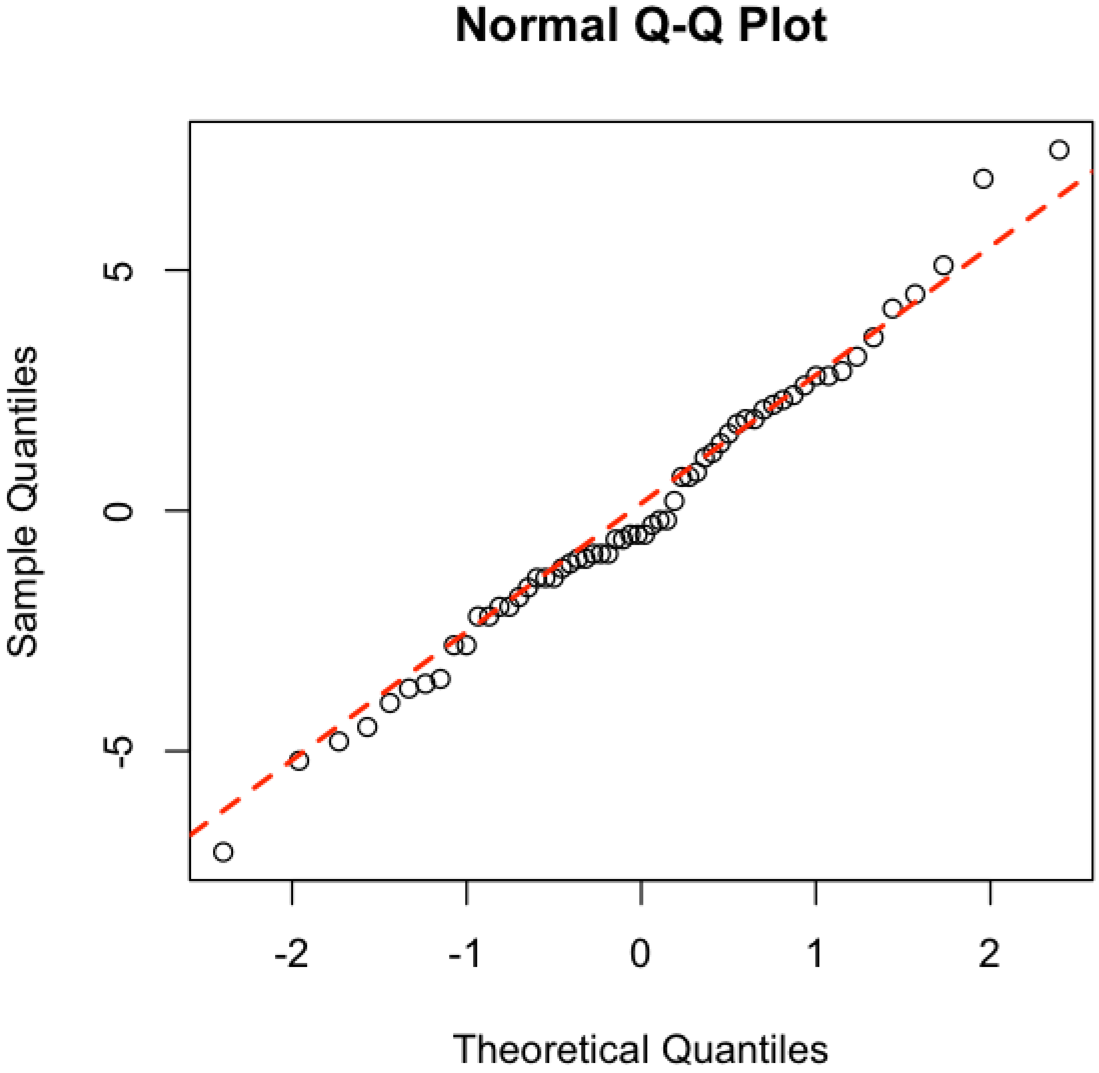
[10]



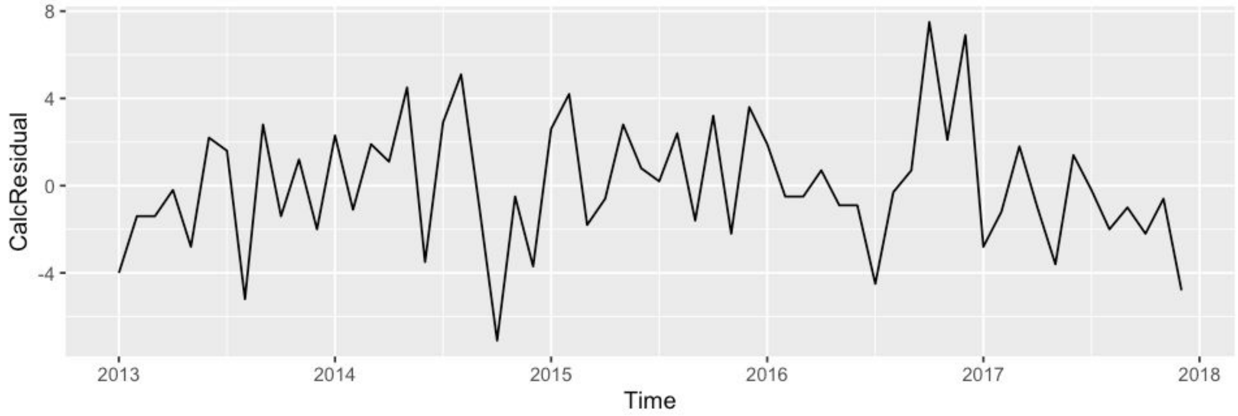
ACF of the residual of Regression model

[11]

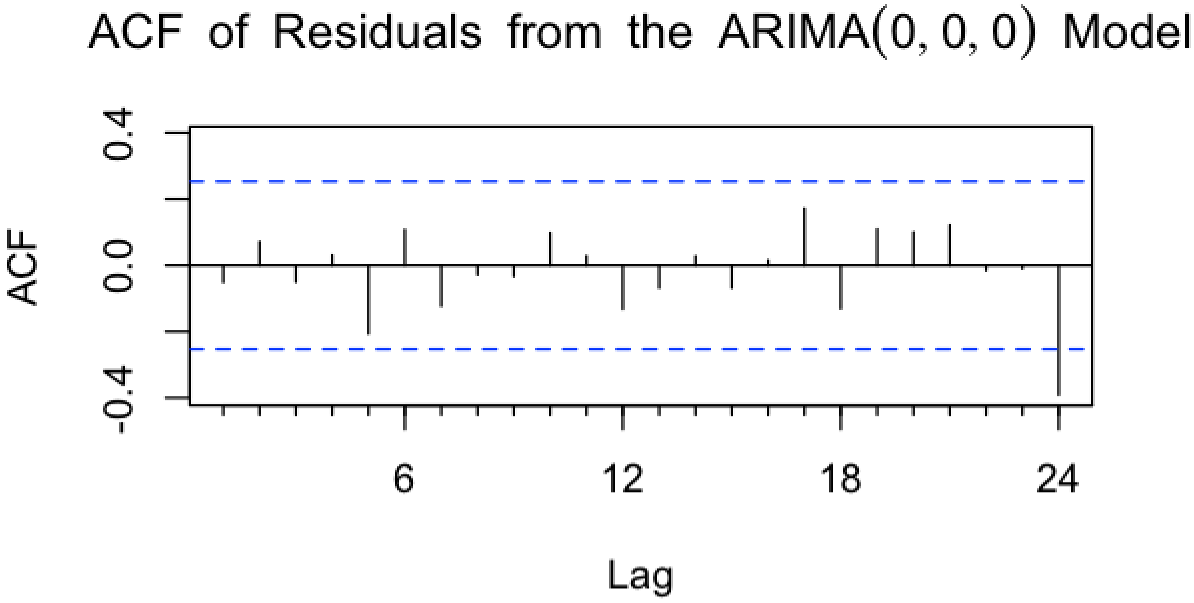

Histogram of ferrors
Density
0.00
0.05
0.10
0.15
0.20
-20
-10
0
10
ferrors


Normal Q-Q Plot
Sample Quantiles
5
0
-5
-2
-1
0
1
2
Theoretical Quantiles

[12]


8
4
0
-4
CalcResidual
2013
2014
2015
2016
2017
2018
Time

[13]


ACF of Residuals from the ARIMA(0, 0, 0) Model
ACF
0.4
0.0
-0.4
6
12
18
24
Lag