Demand Forecasting for a retail company (Kaggle Competition)

Takeshi Oda

2018/08/25

Summary

This is an individual project regarding prediction of sales demand in retail business. I applied exploratory data analysis, built timeseries model and generated sales forcast for next three months in 10 different stores. The challenge which I delt with was publicly open in Kaggle as 'Store Item Demand Forecasting Challenge'.

https://www.kaggle.com/c/demand-forecasting-kernels-only

Dataset

- train.csv
- test.csv

(https://www.kaggle.com/c/demand-forecasting-kernels-only/data)

```
## Warning: package 'dplyr' was built under R version 3.4.4

## 
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
## 
##     filter, lag

## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union

## Warning: package 'forecast' was built under R version 3.4.4

## Warning: package 'ggplot2' was built under R version 3.4.4
```

Data Loading and data cleansing

Data loading

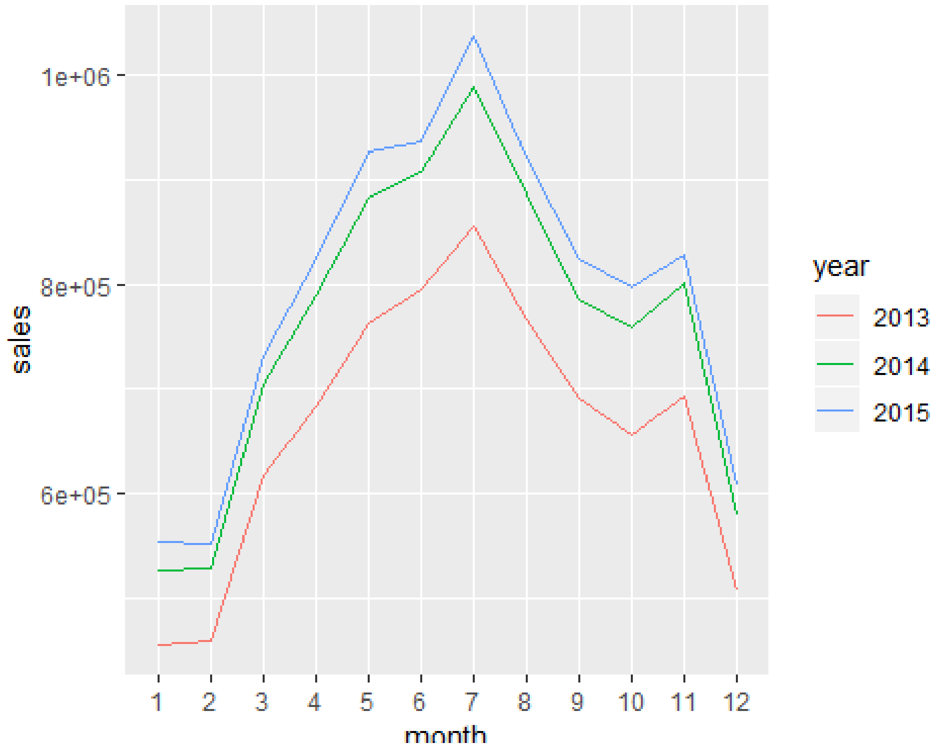
Training data was loaded. In this data, sales amount of 10 items in 50 stores are recorded at daily basis. Since we can see 500 records per day in the training set, there seems to be no missing observation in training set.

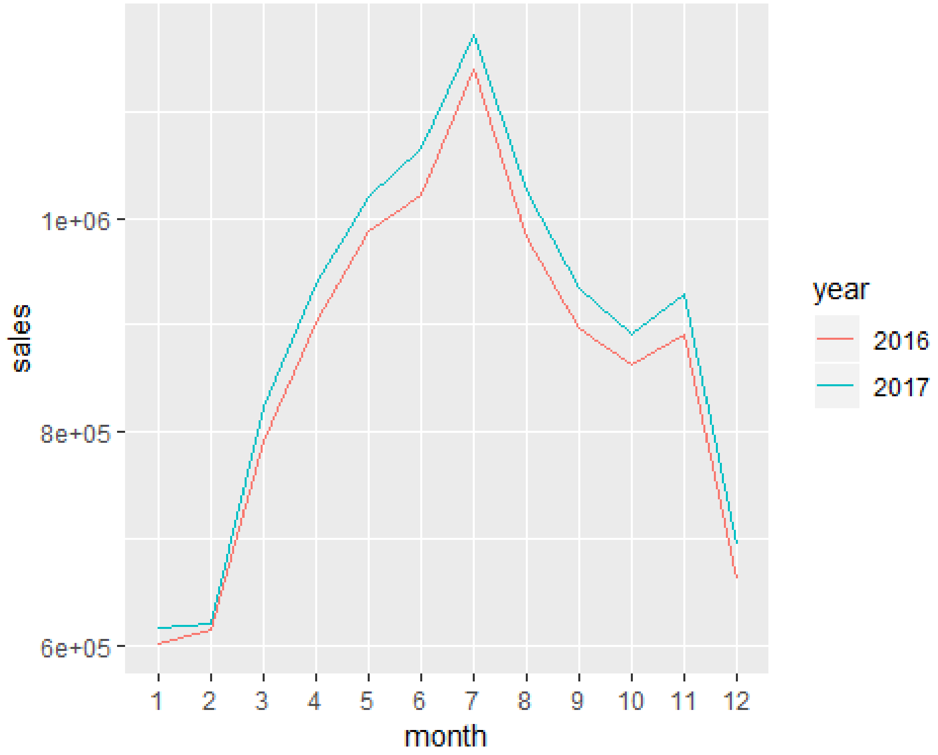
```
## Warning: package 'bindrcpp' was built under R version 3.4.4

## [1] "Number of stores:10"

## [1] "Number of items:50"

## # A tibble: 10 x 2
##    date         cnt
##    <fct>      <int>
##  1 2013-01-01   500
##  2 2013-01-02   500
##  3 2013-01-03   500
##  4 2013-01-04   500
##  5 2013-01-05   500
##  6 2013-01-06   500
##  7 2013-01-07   500
##  8 2013-01-08   500
##  9 2013-01-09   500
## 10 2013-01-10   500

## # A tibble: 10 x 2
##    date         cnt
##    <fct>      <int>
##  1 2017-12-22   500
##  2 2017-12-23   500
##  3 2017-12-24   500
##  4 2017-12-25   500
##  5 2017-12-26   500
##  6 2017-12-27   500
##  7 2017-12-28   500
##  8 2017-12-29   500
##  9 2017-12-30   500
## 10 2017-12-31   500
```

Exploratory Data Analysis

Before building model, we conducted exploratory data analysis. First of all, we take a look at overall trend and seasonality of sales data at all items and all stores.

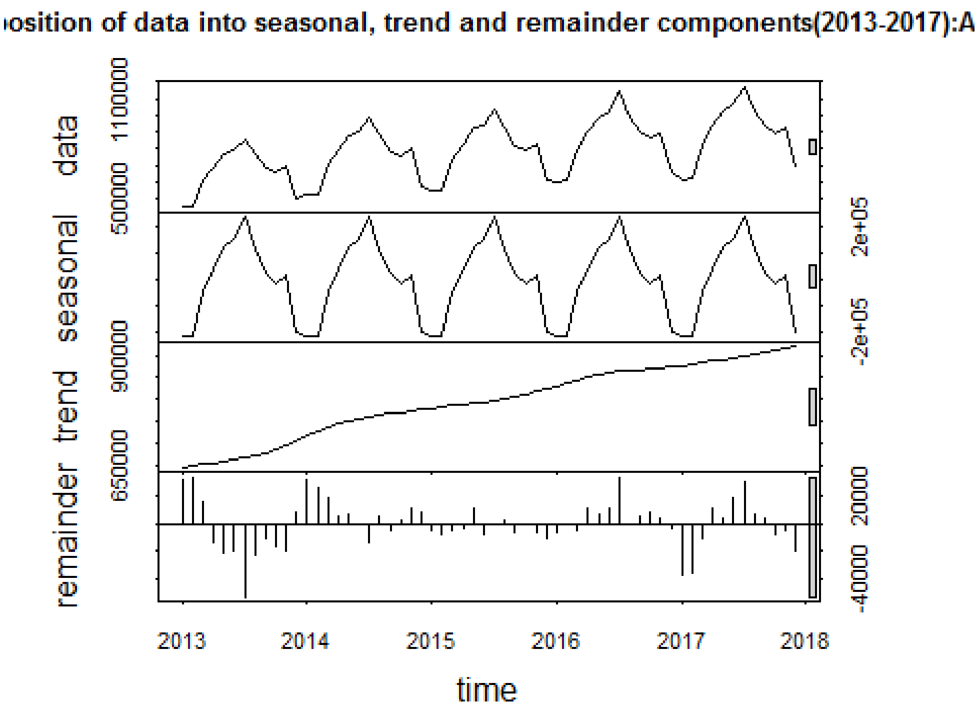
Plot monthly sales at all level


1e+06
8e+05
6e+05
sales
1
2
3
4
5
6
7
8
9
10
11
12
month
year
2013
2014
2015

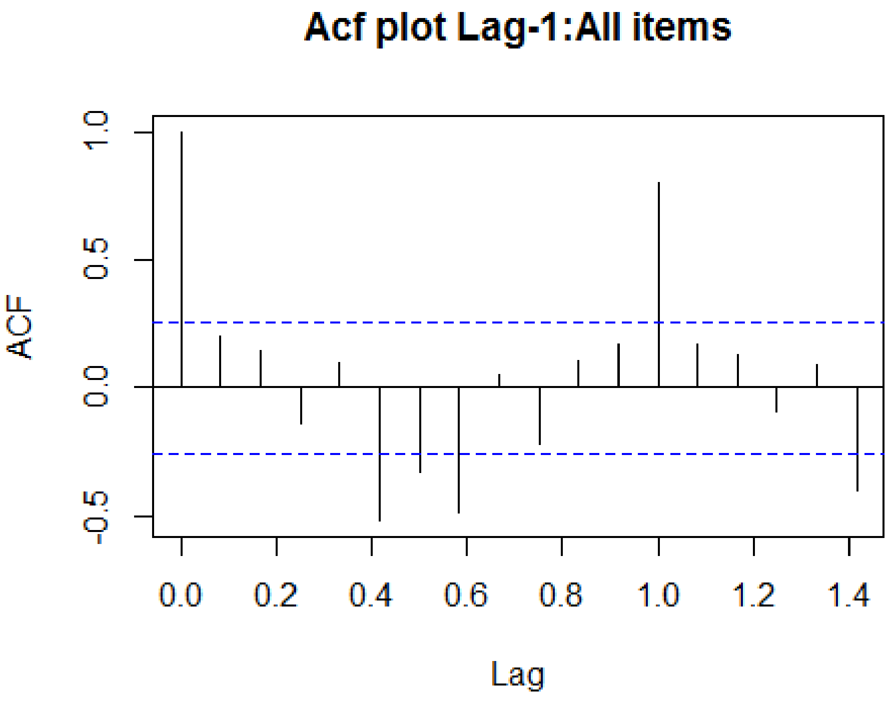

1e+06
8e+05
6e+05
sales
1
2
3
4
5
6
7
8
9
10
11
12
month
year
2016
2017

Plot monthly sales by all stores and all items

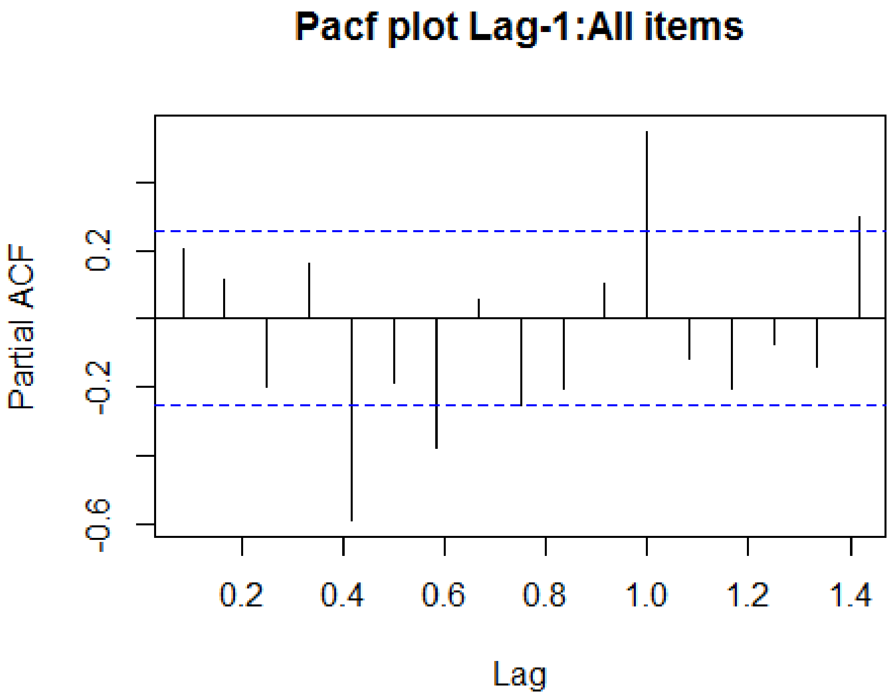
A decomposition of time series data is presented in below. As this chart shows, there is a upward trend in sales for three years. Additinally, there is a seasonal movement of sales. Auto correlation plot tells us that sales data at a month has positive correlation with sales 10 months later having more than 0.5 of correlation coefficient.



osition of data into seasonal, trend and remainder components(2013-2017):A



Acf plot Lag-1:All items

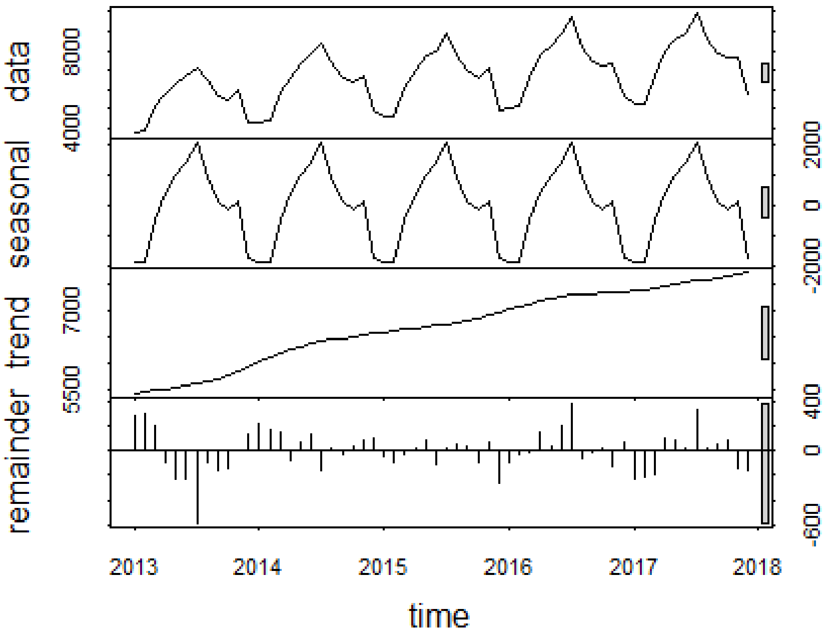


Plot monthly sales by item and all stores

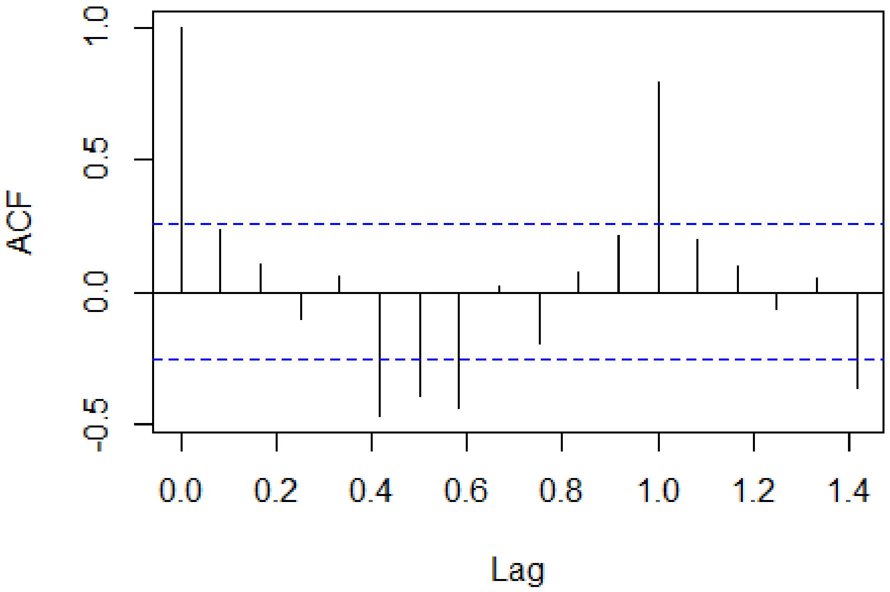
Next, we are going to ask the question whether same trend is found in all items. We split monthly sales data into sections grouped by 10 items and then, apply decomposion process on each subset. Though I have confirmed decomposition of all items, I will present the first three items among them to reduce spaces.

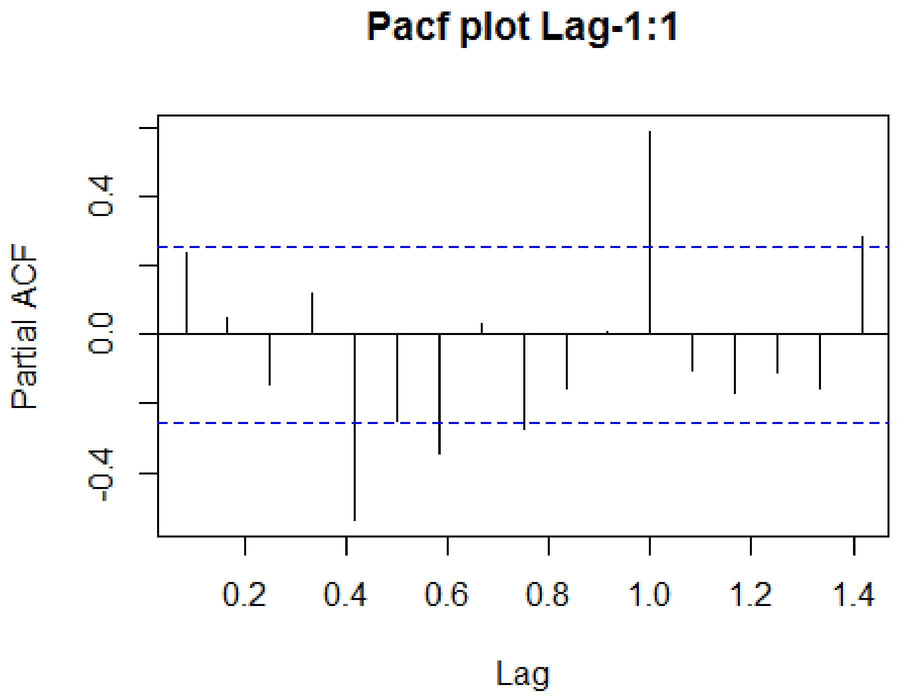
```
items <- unique(data$item)
for (i in 1:length(items)){
        if (i < 3) {
        item_sales <- subset(data, data$item == items[i])
        plot_ts(item_sales, items[i])
        }
}
```

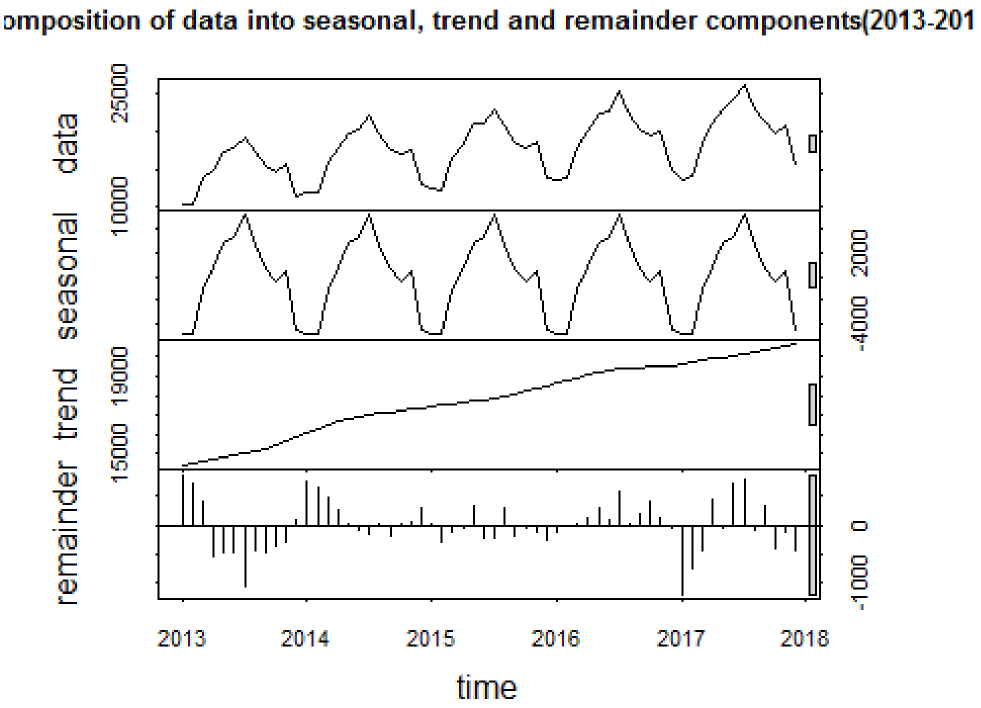
omposition of data into seasonal, trend and remainder components(2013-201

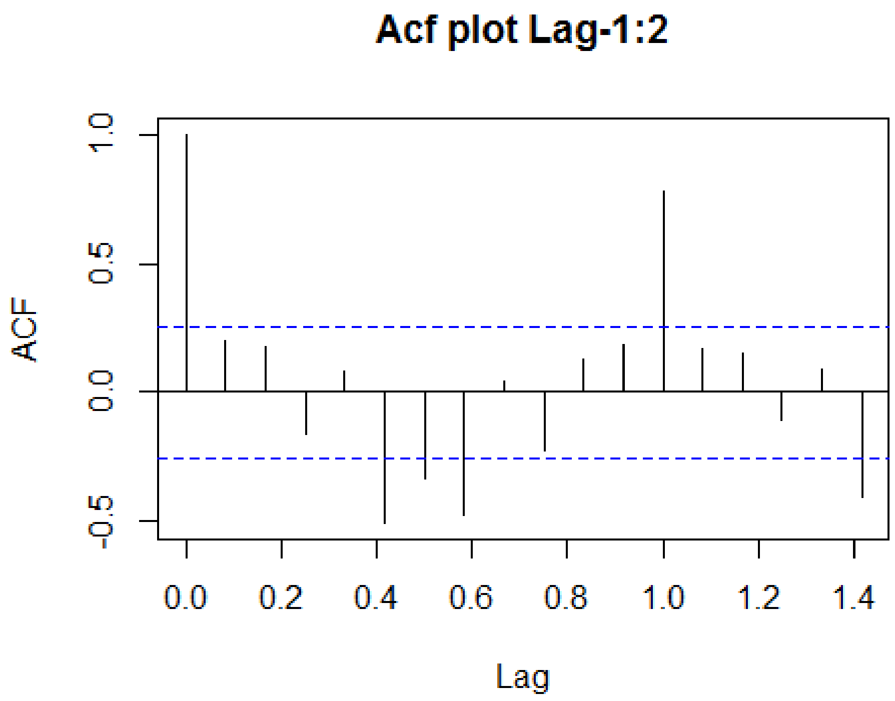


Acf plot Lag-1:1

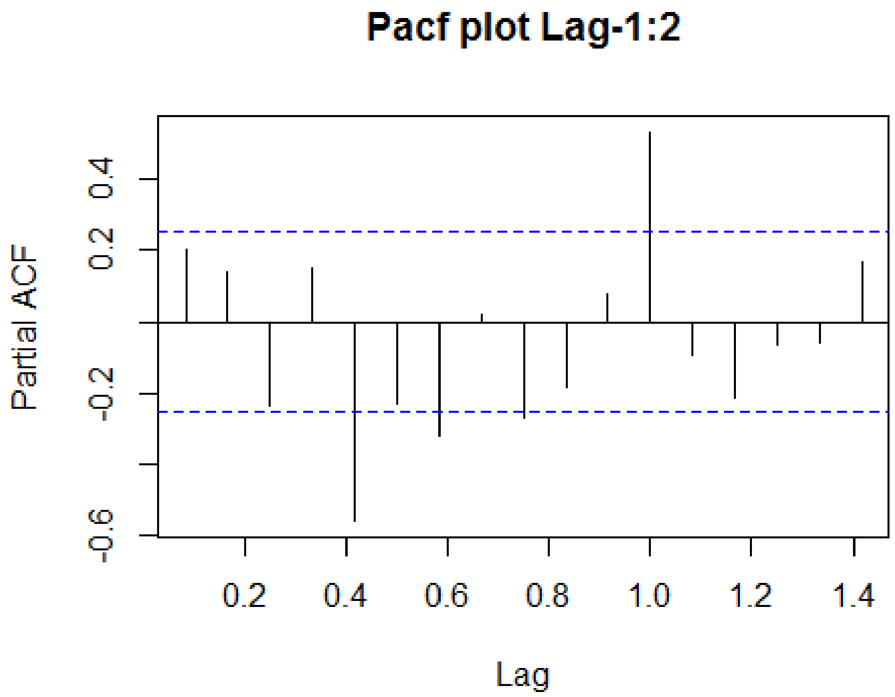



Pacf plot Lag-1:1
Partial ACF
0.4
0.0
-0.4
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Lag


omposition of data into seasonal, trend and remainder components(2013-201
data
25000
10000
seasonal
2000
-4000
trend
19000
15000
remainder
0
-1000
2013
2014
2015
2016
2017
2018
time



Acf plot Lag-1:2



Pacf plot Lag-1:2

##Modeling
Strategy Though we could build time series model on sales data at each item and

store, that approach might involve huge tasks to maintain models and risk of overfitting due to sparsity of data. In addition, we observed that time series data for each item is following almost same trend and cycle. Therefore, we will apply the same model to each combination of item and store. In below, ARIMA model and Exponential Smoothing were built and those performances are displayed.

Build Time Series forecasting model

ARIMA model

Through analyzing autocorrelation plot of time series with 1 degree deffing, we would set parater for ARIMA model as follows.

- p=10
- d=1
- q=10

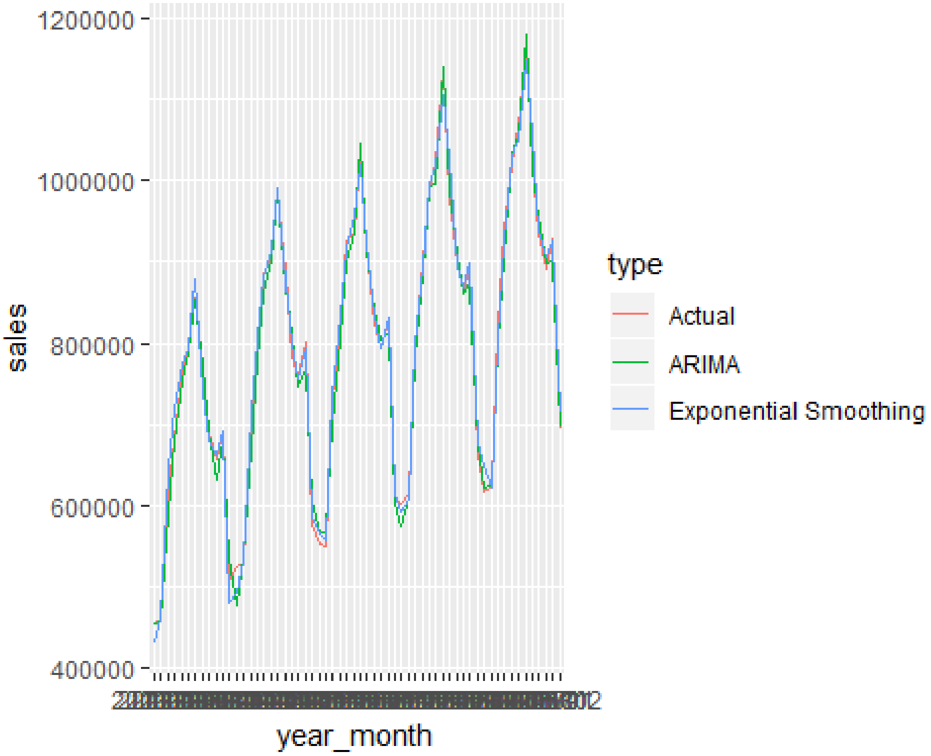
Exponential Smoothing Model

Since this time series forms trend and seasonality, we will apply triple exponential smoothing to sales data.

Comparison of model performance

Exponential smoothing gave better performance metrics.

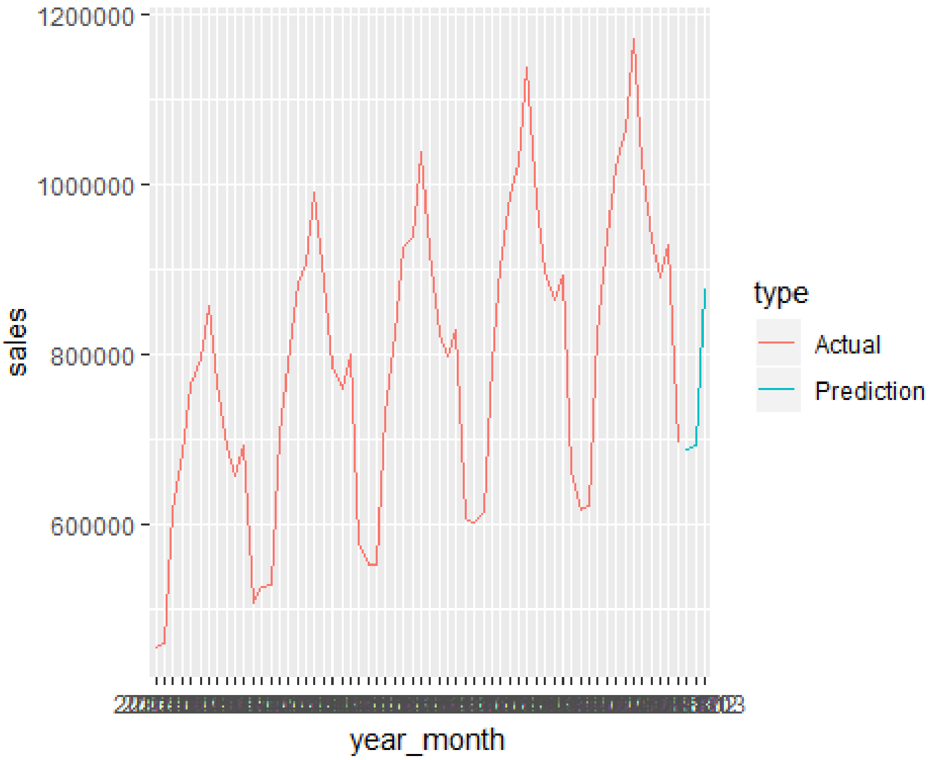
```
##             ARIMA Exponential Smoothing
## RMSE 17591.276867          16238.757361
## MAPE     1.883416              1.665644
```



Prediction

Since we built forecasting model on monthly aggregates of sales data, we will apply same model to generate prediction of sales for each combination of store and item. After gaining predicted sales for next three months, we will break them into prediction at daily level.

We will calculate proportion of daily sales over total monthly sales from training set and the ratio and monthly prediction.



Conclusion

We applied two traditional time series models, i.e. ARIMA model and Exponential Smoothing to predict sales in next three months. Although both model nicely fitted with sales data, we chose Exponential Smoothing model through quantitative comparison of prediction errors.