**OPIM 5503 Data Analytics Using R**

**Instructor: Ram Gopal**



**Team- Data Geeks**

**Topic – Rossmann Store Sales**

**Devesh Singh, Anubha Pant, Pratarshi Saha, Saurav Shekhar and Rohit Gupta**

**Executive Summary:**

Through this report, we aim to compare models which best predicts the sales of 50 Rossmann stores. The given dataset is a combination of cross-sectional and time series data. Due to this we have chosen three methodologies namely.

* Time Series modelling
* Panel Regression based model
* Tree based modeling.

We observed that tree based models gave the best results followed by panel regression and time series forecasting. We could achieve the RMSE of 870 in Classification and Regression Trees.

* We observed both seasonal and non-Seasonal ARIMA components in Time Series Forecasting Models
  + Non- Seasonal: After applying first order differencing the AR and MA orders were identified as 4 and 1 respectively
  + Seasonal: After applying first order seasonal differencing(weekly) the AR and MA orders were identified as 4 and 4 respectively
  + External Regressors: Indicators such as whether the store was open and whether any promotion was running in the store was also helpful in predicting the store sales
* We used pooled estimators, first difference estimators and fixed effect estimators. Among these three panel regression techniques first difference estimators were giving the best test RMSE (1254) and had best RSquare(0.79) as well.
* We also tried tree based as there is no issue of violation of assumption of models thus, producing the best results

**Business Objective:**

Rossmann operates over 3,000 drug stores in 7 European countries. Daily thousands of customers with different demands and behavior visit Rossmann. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. Reliable sales forecasts enable store managers to create effective staff schedules that increase productivity and motivation.

Our objective in this project is to forecast the sales across all the stores for the time dependent data. We aim to create models including time series forecasting, panel regression and tree based models as discussed above. The sales forecast would be made based upon the results from the best model. Accurate prediction of sales will help managers to stay focused on what’s most important to them: their customers and their teams!

**Data:**

The dataset contains historical sales data for 1,115 Rossmann stores over a period from (01-01-2013 to 07-31-2015) with total of 1,017,209 observations. To avoid computational issues we selected only 50 stores.We have also split our data into train and test sets.

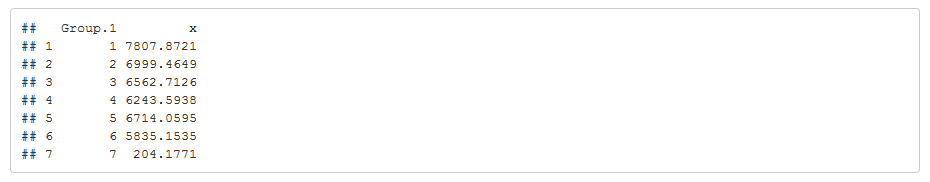
The following are descriptions for the fields in data:

1. Id - an Id that represents a (Store, Date) tuple within the test set
2. Store - a unique Id for each store
3. Sales - the turnover for any given day (this is what you are predicting)
4. Customers - the number of customers on a given day
5. Open - an indicator for whether the store was open: 0 = closed, 1 = open
6. StateHoliday - indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = Public holiday, b = Easter holiday, c = Christmas, 0 = None
7. SchoolHoliday - indicates if the (Store, Date) was affected by the closure of public schools
8. StoreType - differentiates between 4 different store models: a, b, c, d
9. Assortment - describes an assortment level: a = basic, b = extra, c = extended
10. CompetitionDistance - distance in meters to the nearest competitor store
11. CompetitionOpenSince[Month/Year] - gives the approximate year and month of the time the nearest competitor was opened
12. Promo - indicates whether a store is running a promo on that day
13. Promo2 - Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
14. Promo2Since[Year/Week] - describes the year and the calendar week when the store started participating in Promo2
15. PromoInterval - describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store

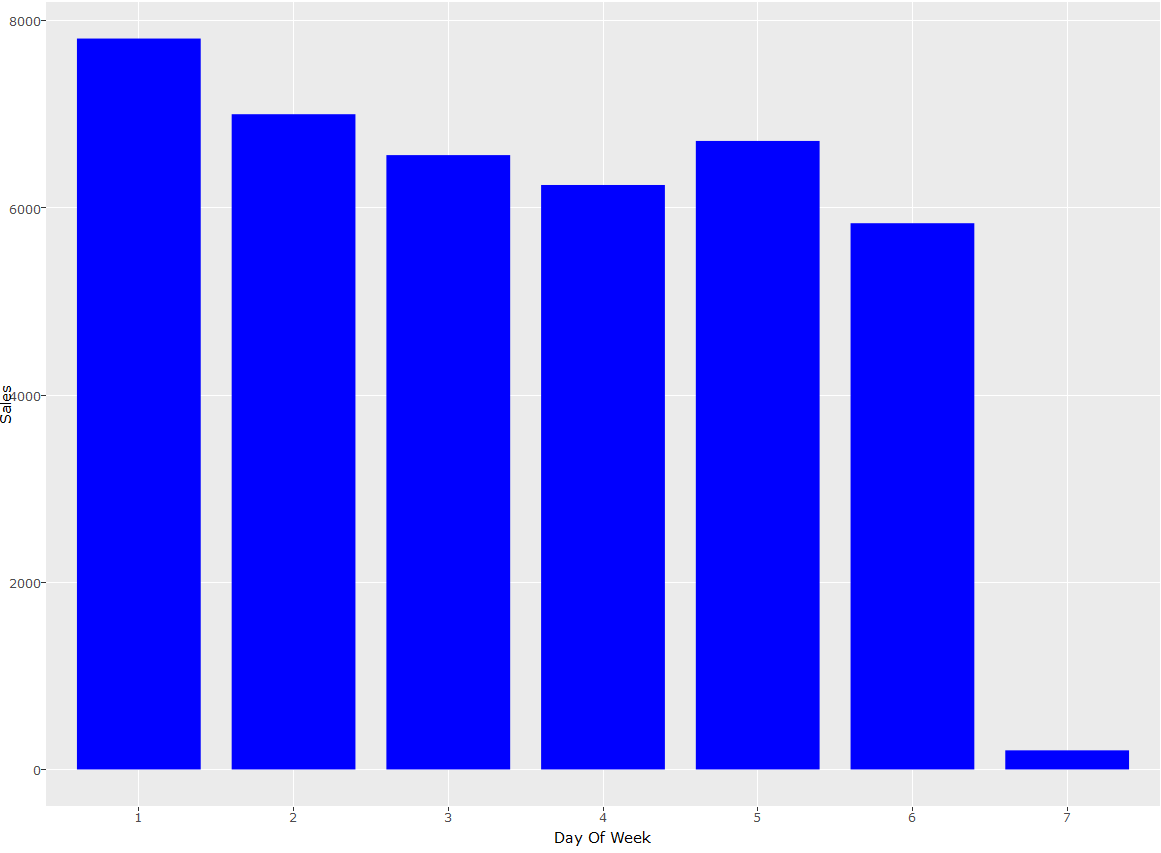
**Exploratory Analysis:**

**Variable**: Day of Week: As the name suggests indicates Day of the Week when sales was made. As seen in the graph Sunday reports the lowest mean Sale and Monday as highest. The distribution of Sales across the week is similar. This can be verified through frequency lot as well as the frequency table listed below.

Frequency Table for DayOfWeek versus Sales



Plotting Sales by DayOfWeek reveals that Sunday has the minimum average sales for the week and Monday has the highest.

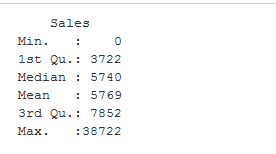


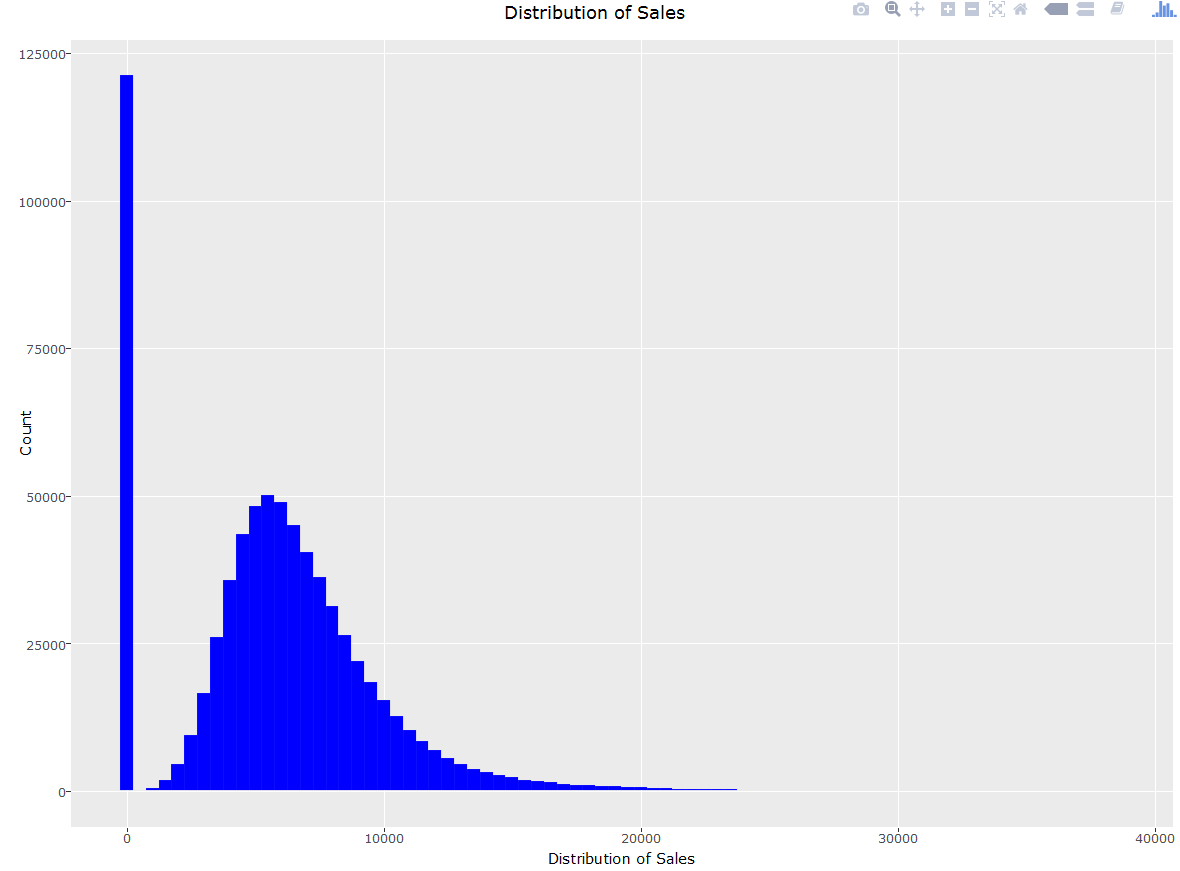
**Variable*:*** Customers

Since we do not have the future number of customers, to put this variable for predicting sales we need to predict the number of customers as well. Therefore, to keep things simple we have not used this variable in our analysis.

**Variable:** Sales

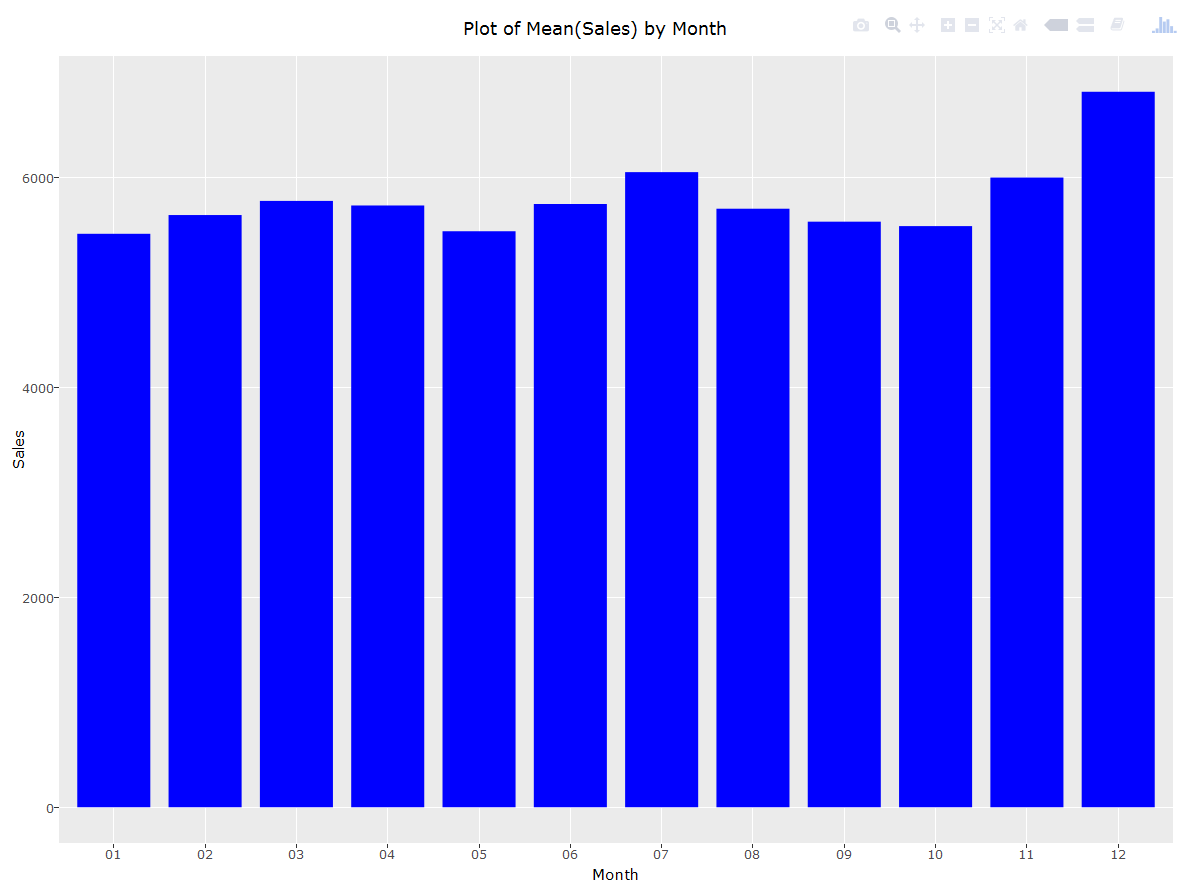
This is our target variable. The frequency distribution states that maximum Sales on a given day was 38722 and there is an average sale of 5769 on a given day.





**Variable:** Month

When we plot mean(Sales) per month , we observed that the sales are slightly higher for December .Since this is a date variable moving forward we will plot seasonal & trend plot to See a better view of how sales is effected for month and year.

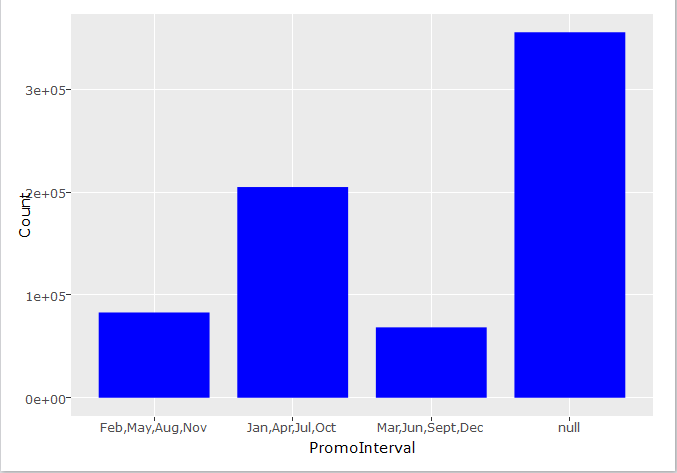


**Variable*:*** Promo2

Promo2 is a variable that indicates whether a store is running a promo on the day of recorded sale. We see that 61% of the time there was no promotional offer while only 38% of the time there was a promotion.

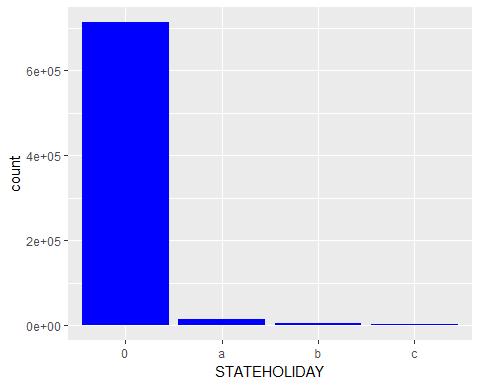
**Variable*:*** PromoInterval

PromoInterval describes the consecutive intervals Promo2 is startedE.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store.Maximum promo2 interval falls in “Jan,Apr,July,Oct” interval. After Plotting the box plots for interval with Mean sales it shows no effect on Sales for different PromoInterval.



**Variable*:*** STATEHOLIDAY

The variable indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None.

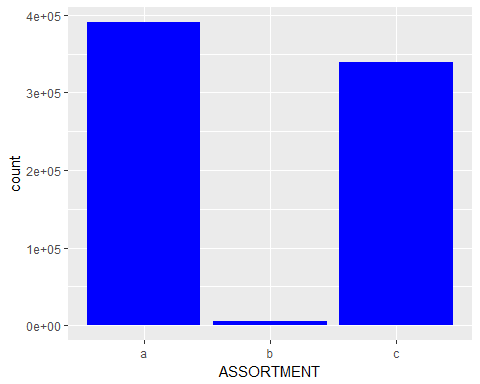


##   
## 0 a b c   
## 96.9751647 1.9571893 0.6409372 0.4267088

The above table shows that almost 97% of the sales happened on days that were not a state holiday and about 3% of the sales were made during a holiday which includes public holidays, Easter holiday and Christmas.

**Variable*:*** ASSORTMENT

The variable describes an assortment level: a = basic, b = extra, c = extended.

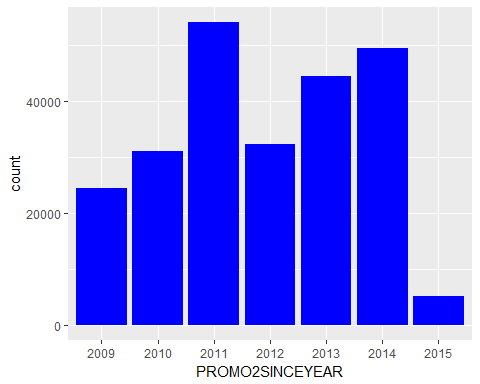


##   
## a b c   
## 53.0983489 0.8167094 46.0849417

The above table shows that assortment of the type 'basic' comprise 53% of the total sales, assortment of type 'extended' comprise of 46% of the sales while 'extra' assortment account for only 0.8%.

**Variable*:*** Promo2Since[Year/Month]

This variable describes the year and calendar week when the store started participating in Promo2.

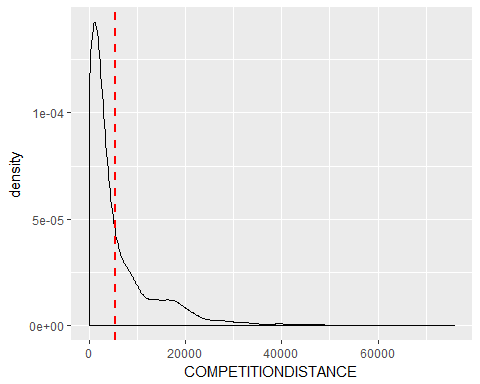


##   
## 2009 2010 2011 2012 2013 2014 2015   
## 12.685232 11.263290 22.409413 14.216757 21.040178 16.635194 1.749936

The above table shows that majority of the sales happened from stores that started Promo2 in the years 2011 and 2013. The stores that started their sales in 2009, 2010, 2012 and 2014 see an equal distribution of sales (11% - 16%) while stores that started Promo2 in 2015 observe only 1.75% of the total sales.

**Variable*:*** COMPETITON DISTANCE COMPETITON DISTANCE

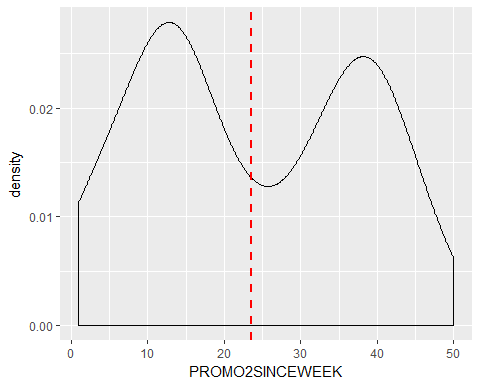
This is the distance in meters to the nearest competitor store



## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 20 720 2320 5397 6880 75860 1985

we can infer that the competition distance is restricted within 20000 meters and the summary of the variable is given below

**Variable**: PROMO2SINCEWEEK



We can infer that 2.7% of sales happened during the PROMO2SINCEWEEK is around 11 and around 2% during week 37. There was a significant dip during the week 23 when the contribution dipped to 1.5%

**Time Series Modeling:**

There are 3 types of time series-

1)**Univariate Time Series**- When we are trying to predict the future value of time series only based upon its past values

2)**Time Series with external regressors**- Predicting future value of time series based upon past values and external regressors

3)**Time Series with dynamic external regressors**- Predicting future value of time series based upon its past values and dynamic external regressors, i.e. the external regressors are treated as time series variables too, which means the past values of external regressors can play a significant role in determining the future value of sales.

There are many models for time series prediction –

1. **Linear Models**- Decomposing the time series into linear trend and seasonal dummies. This is an old technique and now, we have much more advanced techniques
2. **Holt Winters**- This method uses exponential smoothening to capture the trend, while there are multiple ways to capture seasonality
3. **ARIMA models**- Autoregressive integrated moving average models. ARIMA is a forecasting technique that projects the future values of a series based on its own past values plus external regressors.

We will be focusing upon ARIMA time series modeling.

**ARIMA**

The basic assumption of ARIMA time series modeling is Stationarity:

There are three basic criterion for a series to be classified as stationary series :

1. The mean of the series should not be a function of time rather should be a constant.
2. The variance of the series should not a be a function of time. This property is known as homoscedasticity.
3. The covariance of the i th term and the (i + m) th term should not be a function of time.

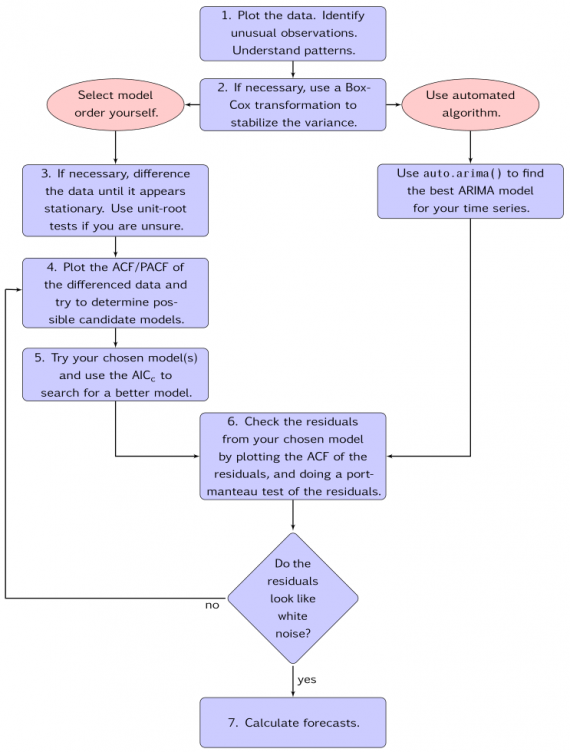
So, these ARIMA models should be built on a stationary time series. If the series is not stationary, we will apply appropriate detrending/DE seasonality techniques to remove trend or seasonality.

After a time-series has been stationarized by differencing, the next step in fitting an ARIMA model is to determine whether AR or MA terms are needed to correct any autocorrelation that remains in the differenced series.

**Missing Value Imputation**

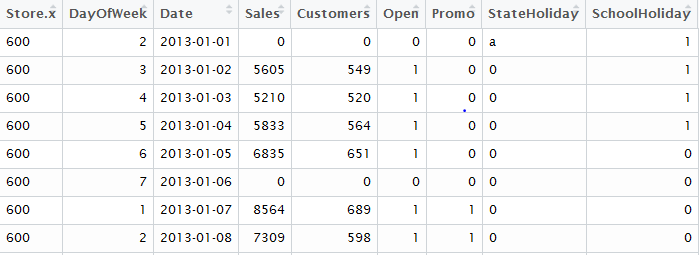
We not do have daily sales data for about 10 stores, out of 50 stores from July-2014 to Dec-2014.  
Since, time series is based upon continuous time intervals, we impute this data first.

**ARIMA procedure**



Note that we will have different time series for each of the 50 stores. For example, the future value of Sales of Store A cannot be predicted using past Sales of Store B. It can only be done using past sales of Store A only. Therefore, we have to build different time series models for different stores. For now, let’s start with analyzing the data for Store Nbr. 600.

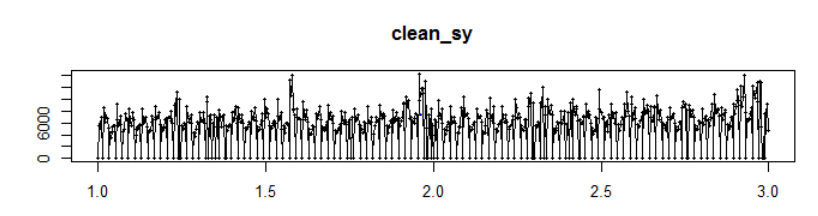
**Data Snapshot**



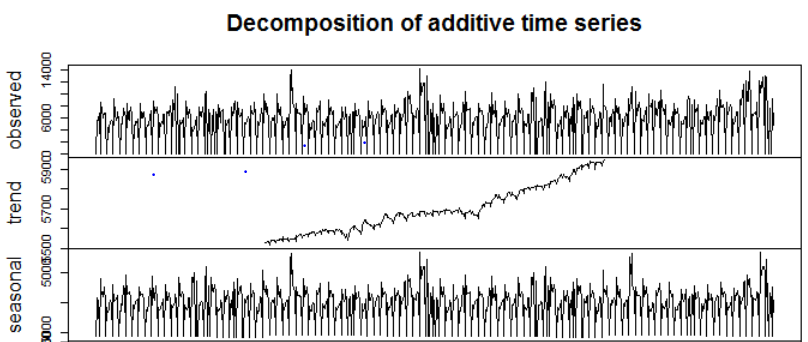
1.We plot the data

sy <- ts(test$Sales,frequency=365)

Frequency=365, for daily level data



A better representation is to use the decompose() in R. This will decompose the time series into possible trend and seasonality components.



There is a possible trend and seasonality in the data based upon above graphs. That means the time series is non-stationary. Let’s validate this by Dicky-Fueller test, which tests for stationarity of the time series.

adf.test(sy, alternative = "stationary")

Augmented Dickey-Fuller Test

data: clean\_sy

Dickey-Fuller = -11.658, p-value = 0.21

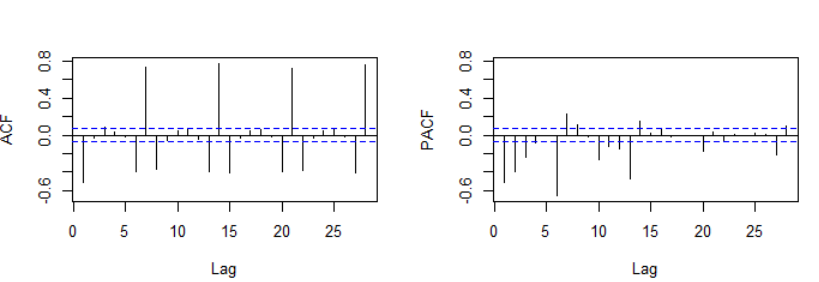
alternative hypothesis: stationary

Since, we cannot reject the null hypothesis, this validates that the time series is non-stationary.

2.If the time series is non-stationary, difference the time series unless it becomes stationary.

Taking the first difference: tsdisplay(diff(sy),lag.max=28)

3.Let’s explore the ACF and PACF plots after 1st order differencing



Seasonality can be seen at Weekly, Weekly-1 and Weekly+1, i.e. spikes at 6,7,8; 13,14,15 and similarly.  
**Non-Seasonal ARIMA**

For non-seasonal ARIMA, the order of AR and MA terms after 1st order differencing, as seen from the graphs are 4 and 1 respectively.

Let’s include these terms and fit a non-seasonal ARIMA model.

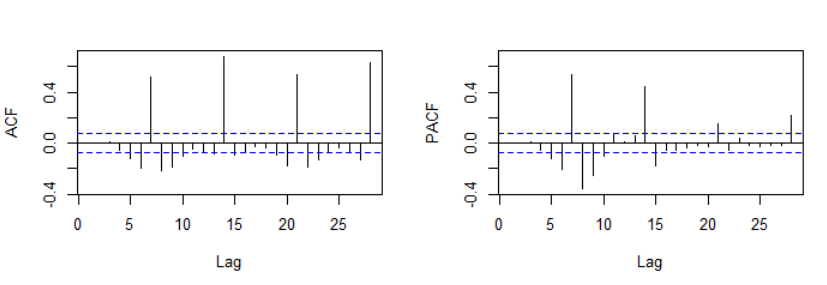
model <- Arima(sy, order=c(4,1,1),seasonal = list(order = c(0, 0, 0)))

4. Use RMSE to select the best model



5. Let’s analyze the residuals now. The residuals should not have any patterns ideally.

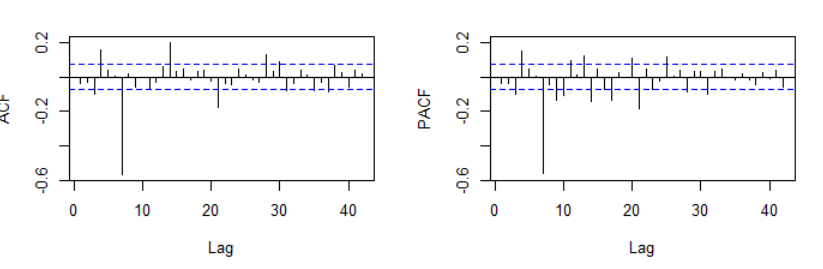
ACF and PACF plots



Note that the significant lags, except the seasonal lags have been removed in the residuals.  
The only significant lags can be observed at weekly, weekly-1 and weekly+1 lags.  
  
**Seasonality**

**Seasonal ARIMA**

1.Apply weekly Seasonal Differencing and explore the ACF and PACF plots



Note that for both ACF and PACF plots, lags at 7,14,21 and 28 days comes out to be significant.  
Since, we have applied weekly seasonal differencing, order 1 =7 days, order 2 = 14 days and so on.  
  
So, the seasonal AR and MA orders are 4.

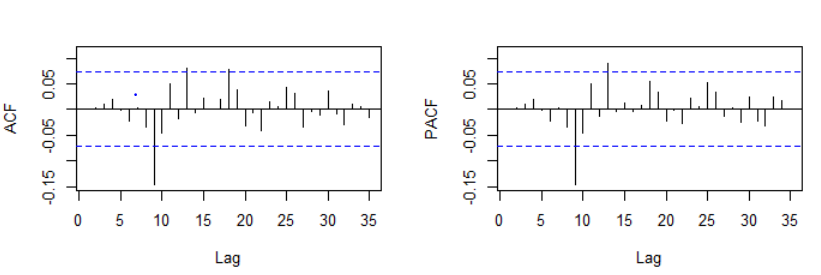
2. Including these into the models

model <- Arima(sy , order=c(4,1,1),seasonal = list(order = c(4, 1, 4),period=7))



Note that the RMSE is decreased to 1667 in the training dataset.

3. Analyzing the residuals



The ACF and PACF plots show spikes at lags 9 and 14, which is very strange.   
Let’s also include the external regressors and try to see their impact

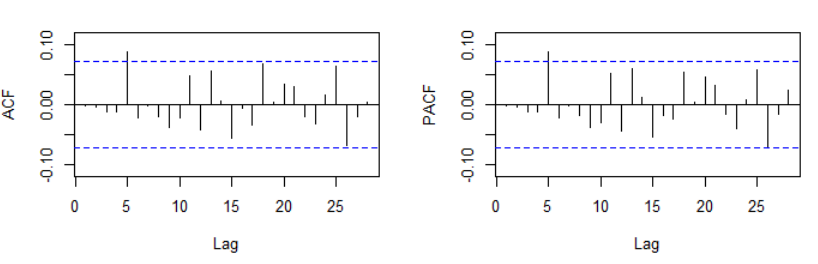
ARIMA with external regressors

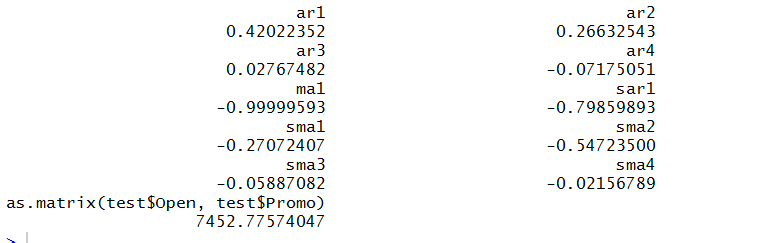
model <- Arima(sy, order=c(4,1,1),seasonal = list(order = c(1, 1, 4),period=7),xreg = cbind(test$Open,test$Promo))



Training RMSE has decreased to 1157.

Analysis the residuals



This seems a lot better. The ACF and PACF plots show that the lags now are more or less insignificant which means that there is no significant autocorrelation present in the data now.  
Variable co-efficient in the final model  


RMSE on test dataset: **1957**  
  
**Additional Models**

**Linear Regression**

Linear regression is the most basic type of regression and commonly used predictive analysis.  The concept of regression is to examine two things:

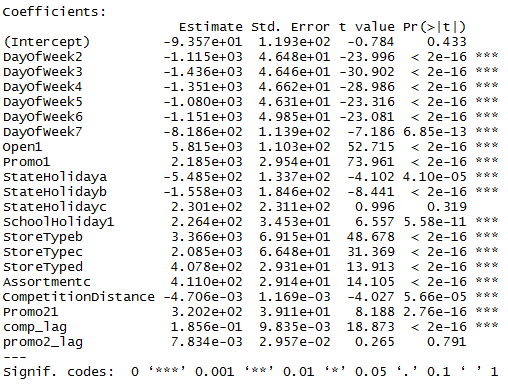
* Whether a set of predictor variables do a good job in estimating the outcome variable and the variability in the dependent variable that is accounted for by the set of predictors.
* Identify the significant predictors of the dependent variable and in what way do they--indicated by the magnitude and sign of the beta estimates--impact the dependent variable.  These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables.
* Finally, identify the regression equation that shows how the set of predictor variables can be used to predict the outcome.

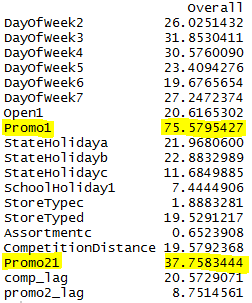
The simplest form of the equation with one dependent and one independent variable is defined by the formula

y = c + b\*x

where y = dependent variable, c = constant, b = regression coefficients, and x = independent variable.

After performing initial data cleaning, we fit the Linear model where Sales column was the dependent variable and the Sales and Store attributes as the independent variables.





Based on the developed model, the variables that are identified as most important are Promo and Promo2. Stores offering promos show increased values in sales as compared to stores not offering promos. However, we will not consider the model for our data since the performance metrics are not substantial. The model has an R-squared value of 0.64 and an RMSE of 1634.451.

**Panel Model:**

Panel data is multi-dimensional data involving measurement of individual entities over a certain period. It is basically combination of cross sectional and time series data. The given data is panel data as it contains information of 1,115 stores captured daily. So, we tried to see if the results can be improved using panel regression.

Intuition behind panel regression

One of the important assumptions of OLS is that the error term should be independent at each predictor value. Another important assumption is that the error terms should be randomly distributed across the mean (constant variance, no autocorrelation). Running simple OLS violates these two assumptions. This leads to biased and instable model. So, to overcome this we use panel regression approach on panel data. Panel regression involves various transformations to make sure that the errors adhere to the assumptions.

**Approaches of Panel Data Estimation**

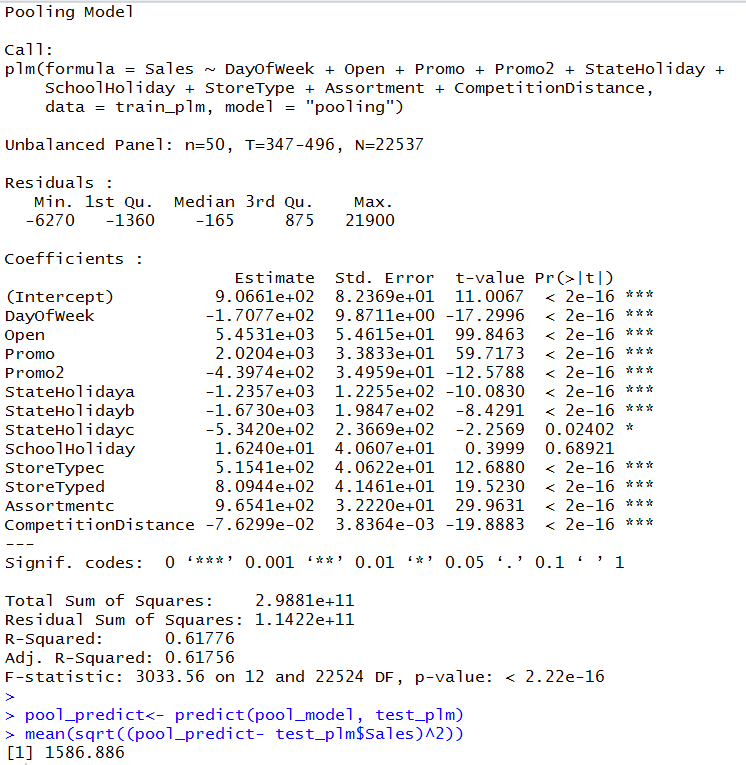
Below are the panel data estimation approaches:

1. Pooled OLS Estimator
2. First Differences Estimator
3. Fixed Effects Estimator
4. Random Effects Estimator

We have used Pooled OLS Estimator, First Difference Estimator and Fixed Effects Estimator.

**Codes and results**

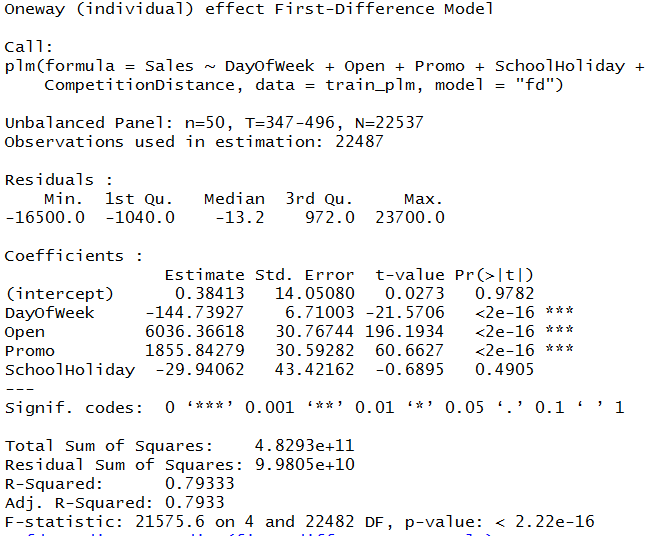
Pooled OLS



**Observation:**

1. RMSE for the model was 1,586 which is better than Time Series
2. Except for “School Holiday” all the variables are significant
3. R-Square for the mode is only 0.61, so there is scope of improvement and this calls for trying other panel regression estimation techniques

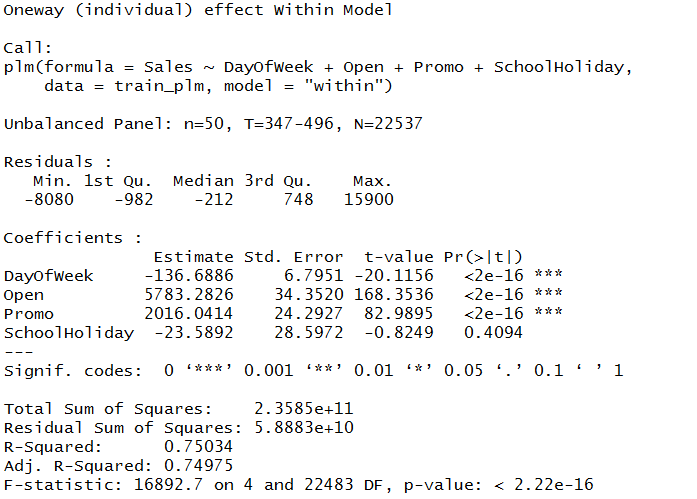
**First Difference Estimator**



**Observation:**

1. RMSE for the model was 1,254 which is better than pooled models
2. Time invariant variables could not be included in First Difference models
3. Despite dropping time invariant variables, the R-Square has improved to 0.79 from 0.61
4. Even this model says that “School Holiday” does not have significant effect on the sales

**Fixed Effect Estimator:**



**Observation:**

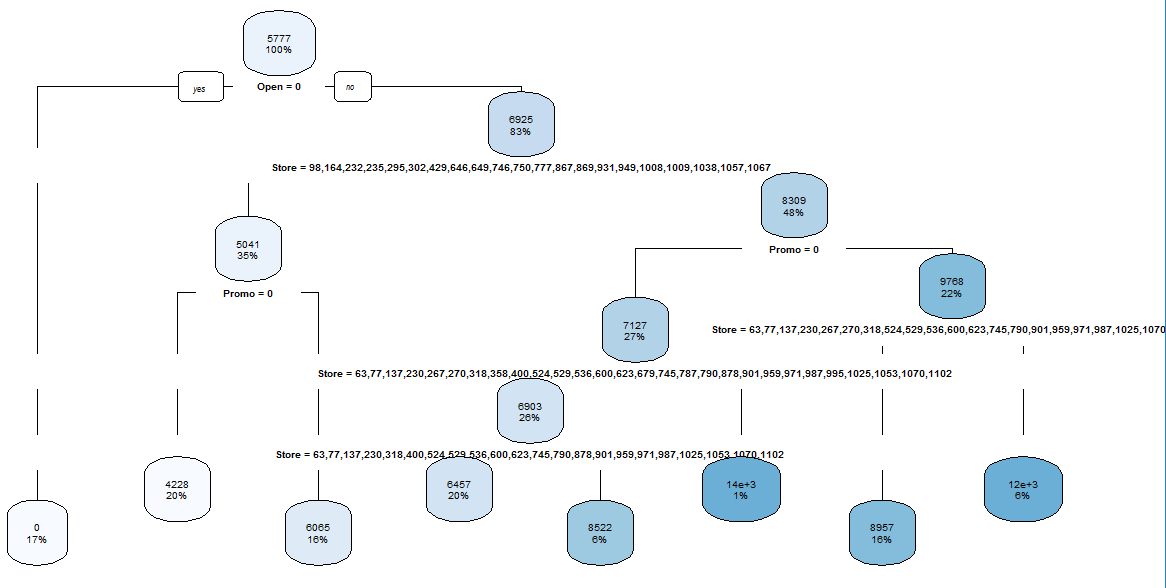
1. Test RMSE for the model was 1,312 which is better than pooled models but not better than First Difference estimators
2. R-Square has also decreased compared to First Difference estimators
3. “School Holiday” does not have significant effect on the sales

**Tree Based Models**

In addition to Panel Regression we used Tree based model to predict the sales of Stores. As we know highly nonlinear relationships between variables will result in failing checks for simple regression models and thus make such models invalid. We then used Non-parametric approach which are not based on any such assumption to forecast the sales for Store. One such modeling techniques is Tree based modeling. We are using Decision trees are a simple, but powerful form of multiple variable analysis. They provide unique capabilities to supplement, complement, and substitute for traditional statistical forms of analysis. Decision trees are produced by algorithms that identify various ways of splitting a data set into branch-like segments. The values in the input field are used to estimate the likely value in the target field. The target field is also called an outcome, response, or dependent field or variable.

We have used following R packages to run Decision Tree model.

* CART is a tree based approach and the split at each node is performed on the decrease of RMSE
* We ran the model to plot the tree and using the plot command. The plot generated will be a very messy and will not give a good visualization.
* Using the package PARTY and RColorBrewer we again plotted the tree with a pruned tree and the visualization seems better than the normal tree plot



Based on the developed model, the variables that are identified as most important is OPEN. The tree then uses the PROMO and STORES itself for further splits. OPEN column signifies that the store was open on the given day or not which is a significant factor. The model has an RMSE value of 870.

Conclusion

|  |  |  |
| --- | --- | --- |
| Model Name | Summary | RMSE |
| ARIMA | ARIMA (4,1,1)  Seasonal ARIMA (4,1,4) | 1957 |
| Panel Regressor Model | First Difference Model, most important variables are Open, Promo, DayofWeek | 1254 |
| CART | Most important variables Open, Store, Promo | 870 |

We observed that store level attributes could not be included in ARIMA models. Also the external regressors could be dynamic i.e. past values of regressors could be helpful in determining future value of store sales. Since, the results were not good thus we proceeded with panel regression.

As discussed above panel regression is performed under certain assumptions. The results got better while performing panel regression but since the assumptions were not met, we tried developing tree based models which do not involve any assumption on the given data. These gave a RMSE of 870 which in fact was the lowest RMSE as expected. ***Thus, tree based models proved to be most efficient.***

***Reference***

* [www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf](http://www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf)
* machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/
* blog.revolutionanalytics.com/.../plotting-classification-and-regression-trees-with-plotr
* [www.statmethods.net/advstats/timeseries.html](http://www.statmethods.net/advstats/timeseries.html)
* a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html
* https://www.analyticsvidhya.com › Business Analytics
* robjhyndman.com/talks/MelbourneRUG.pdf
* Panel Data using R - Princeton University
* https://rpubs.com/wsundstrom/t\_panel

**Appendix**

Packages used

**RPART**

The rpart programs build classification or regression models of a very general structure using a two- stage procedure; the resulting models can be represented as binary trees. The tree is built by the following process: first the single variable is found which best splits the data into two groups. The data is separated, and then this process is applied separately to each sub-group, and so on recursively until the subgroups either reach a minimum size or until no improvement can be made. It works similar way to linear model where we must input the formula.

**RATTLE**

It is a graphical user interface for building decision trees. Rattle adds value to the basic rpart functionality with additional displays of the decision tree, as in Figure 9, and the conversion of the decision tree into a list of rules (using the Draw and Rules buttons respectively). Time series analysis is not directly supported in Rattle. This package provides a straight-forward interface to a collection of descriptive and predictive model builders available in R. For each, a simple collection of tuning parameters is exposed through the graphical interface

**RPART.PLOT**

An Enhanced Version of 'plot.rpart'. It will help to plot the decision tree generated by the RPART.

Use the tweak argument to make the text larger, e.g. tweak=1.2. This may cause overlapping labels. However, there is a little elbow room because of the whitespace between the labels.

**PARTY**

This package does the recursive partitioning for building tree-based regression and classification models. This includes conditional inference trees, conditional inference forests and parametric model trees.The core of the package is ctree. It embeds tree-structured regression models into a well-defined theory of conditional inference procedures. Party package also uses the framework of graphical appearance control and provides several panel generating functions.

**PARTYKIT**

A toolkit with infrastructure for representing, summarizing, and visualizing tree-structured regression and classification models. This unified infrastructure can be used for reading/coercing tree models from different sources such as RPART etc., yielding objects that share functionality for print ()/plot ()/predict () methods. For querying the dimensions of the tree, three basic functions are available: length () gives the number of kid nodes of the root node, depth () the depth of the tree and width () the number of terminal nodes. The usage is not only restricted to visualizations but it can also show good statistical graphics.

**FORECAST**

Methods and tools for displaying and analyzing univariate time series forecasts including exponential smoothing via state space models and automatic ARIMA modelling

**TSERIES**

Time series analysis and computational finance