**PROJECT TOPIC - STORE SALES DEMAND FORECASTING**

The objective of our assignment is to predict three months sales values for different stores for the year 2018 that is for the months of January, February and March, given the day to day data from 2013 to 2017 for 10 stores each selling 50 identical item`s.

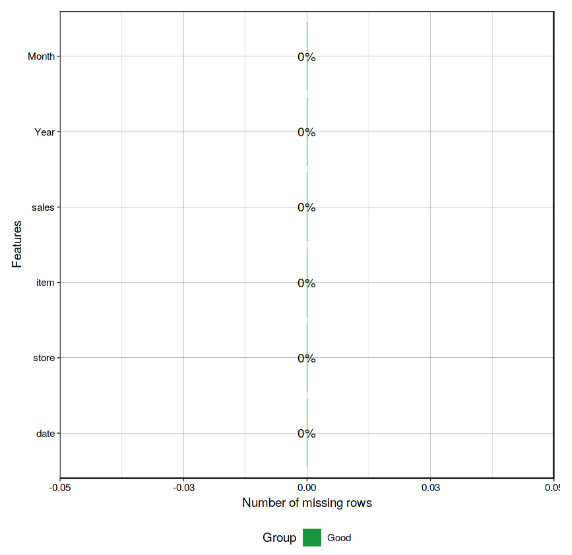
Data is recorded everyday for 1826 days for 10 stores for 50 items which makes total number of observations equal to 913000 (=1826\*10\*50). So the structure of the given data is panel data although time series analysis can be done for panel data with some modifications but we decided to go for pure time series analysis for this we divided the data on the basis of stores and items resulting in 500 time series data. We created a prediction function to which we passed all these 500 time series to make sales prediction. The first task at hand was to check the data quality, pre-process the data and visualising the data to make sure the data is of good quality and can be used for the given task.

**Data Pre-processing and Visualization**

The relevant parameters for checking data quality are:

1. Accuracy (no typos or junk values)

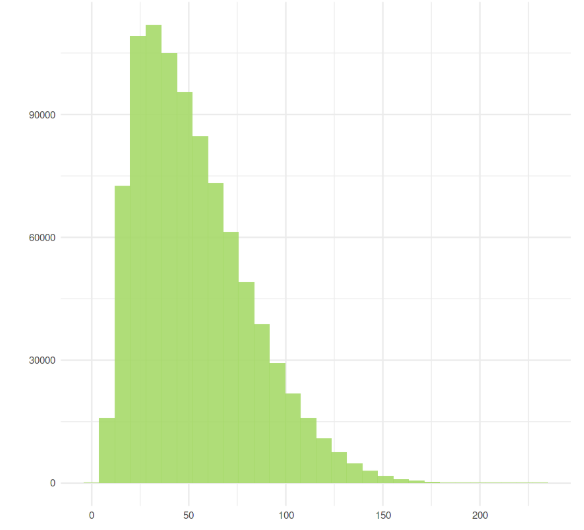
2. Completeness (no missing vales or truncation)

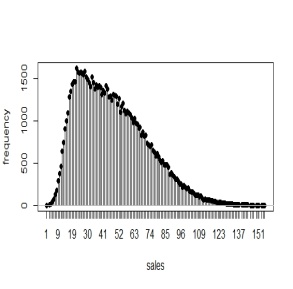
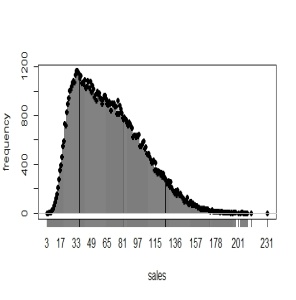
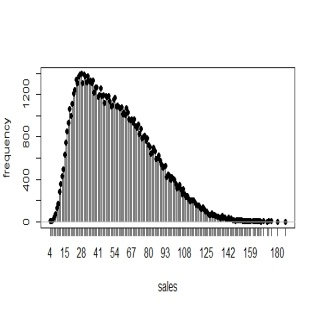
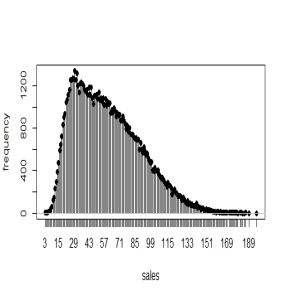


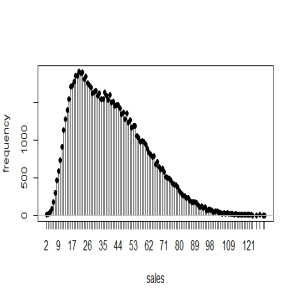
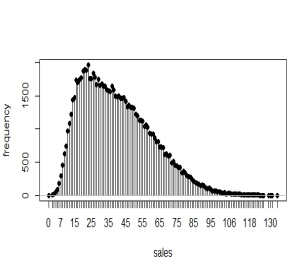
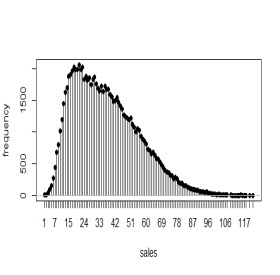
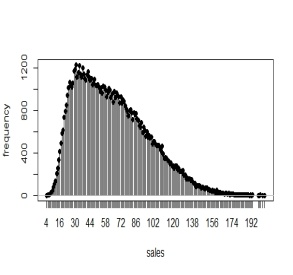
3. Consistency (no contradiction)

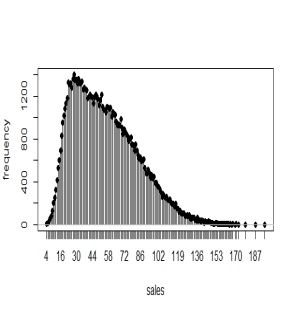
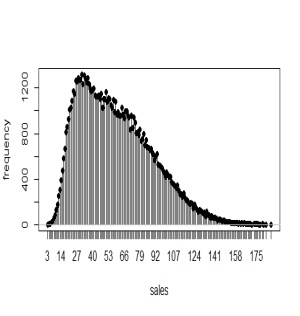
4. Validity (allowable values)

From the analysis we found all the attributes in the data followed all the above data quality measures making it fit for analysis. Also we found that we cannot remove the outliers because they are mostly due to holidays and holidays improved the fit of the model. We visualized the data based on stores, items and time series analysis on daily weekly monthly and yearly basis and trend in the sales values with the years. We found that the train data followed **Positively Skewed Distribution** for sales, followed same distribution for different stores, different items and for different years.



 **(Sales Distribution for different stores)**

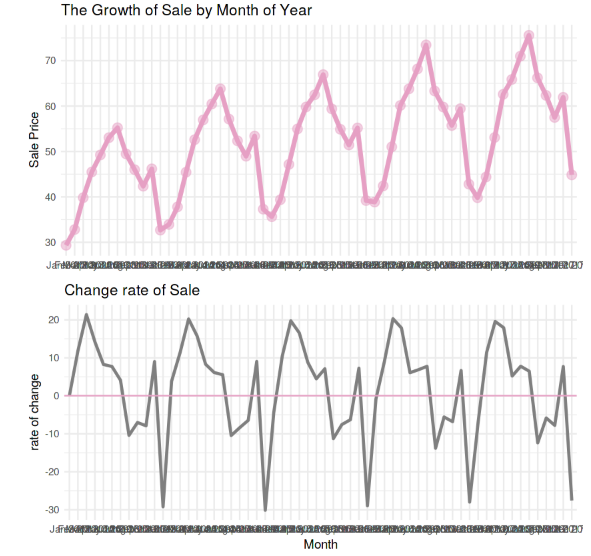
**Time series trends**

**Daily**: Sales is fluctuating in the day time and starts rising in the evening and through the night.

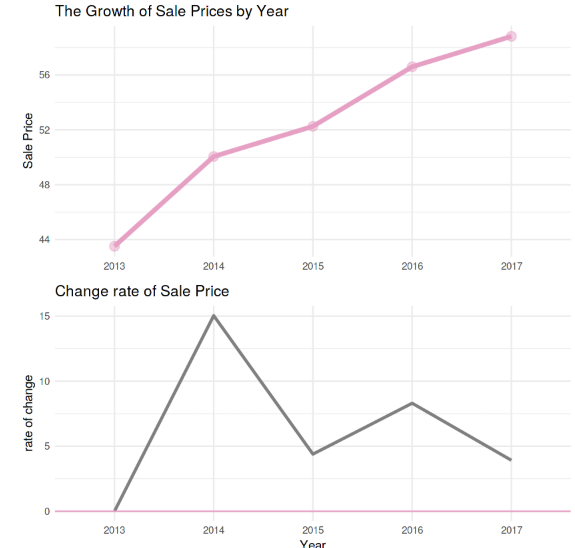


**Weekly:** The sales were highest on Sundays and drop to lowest on Mondays and then increases steadily through the week.

**Monthly**: The sales were low in the starting month of the year and then rises from February and steadily increasing and reaching maximum in July and then decreases till October and remains almost constant in November and rises sharply in December due to festive occasions like Christmas and New Year.

****

**Yearly**: There is a steady increase in sales with year.



**Model used:**

At its core, the **Prophet Procedure** is an additive regression model with four main components: A piecewise linear or logistic growth curve trend. Prophet automatically detects changes in trends by selecting change points from the data. A yearly seasonal component modelled using Fourier series.

**Yhat=trend\*(1+multliplicative terms) +additive terms**.

A). The cool thing about Prophet is that it doesn’t require much prior knowledge or experience of forecasting time series data since it automatically finds seasonal trends beneath the data and offers a set of ‘easy to understand’ parameters. Hence, it allows non-statisticians to start using it and get reasonably good results that are often equal or sometimes even better than the ones produced by the experts.

B). Prophet modelling can be able to detect the Change Points in time series data.

C) We can include the holidays (Play-offs & Super-bowls) in our data. Details have been added later.

D.) We can regularise the parameters by means of Bayesian optimisation with cross-validation.

E) We can incorporate the multiplicative-seasonality and determine the uncertainty intervals in the data.

F) Additional regressors can be added to the linear part of the model using the **add\_regressor** method or function. A column with the regressor value will need to be present in both the fitting and prediction data-frames.

**Seasonality and Additional Regressors**

Seasonality is estimated using partial Fourier sum which can approximate an arbitrary periodic signal. The no of terms in the partial sum (the order) is a parameter that determines how quickly the seasonality can change. Additional regressors can be added to the linear part of the model using the add\_regressor method or function. A column with the regressor value will need to be present in both the fitting and prediction data frames. We have added an additional effect on Sundays during the **NFL season (National Football League)**. Extra regressors are put in the linear component of the model, so the underlying model is that the time series depends on the extra regressor as either an additive or multiplicative factor.

**Effects of holidays**

If you have holidays or other recurring events that you’d like to model, you must create a data frame for them. It has two columns (holiday and ds) and a row for each occurrence of the holiday. It must include all occurrences of the holiday, both in the past (back as far as the historical data go) and in the future (out as far as the forecast is being made). If they won’t repeat in the future, Prophet will model them and then not include them in the forecast. You can also include columns lower\_window and upper\_window which extend the holiday out to [lower\_window, upper\_window] days around the date. For instance, if you wanted to included Christmas Eve in addition to Christmas you’d include lower\_window=-1, upper\_window=0. If you wanted to use Black Friday in addition to Thanksgiving, you’d include lower\_window=0, upper\_window=1. You can also include a column prior\_scale to set the prior scale separately for each holiday, as described below.

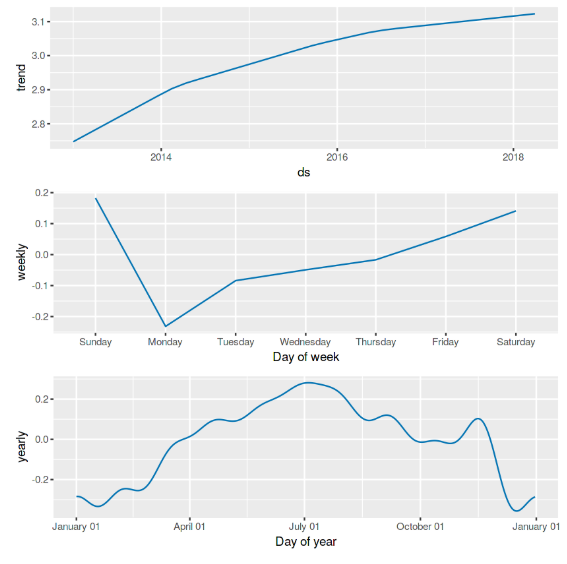
There are **2** types of Holidays:

1. **Playoffs**: These are the Less-Important public holidays and weekends.
2. **Super bowls**: These are the Festive Holidays with high importance. There may occur a high increase or decrease in sales due to these holidays e.g. New Year, Christmas etc.

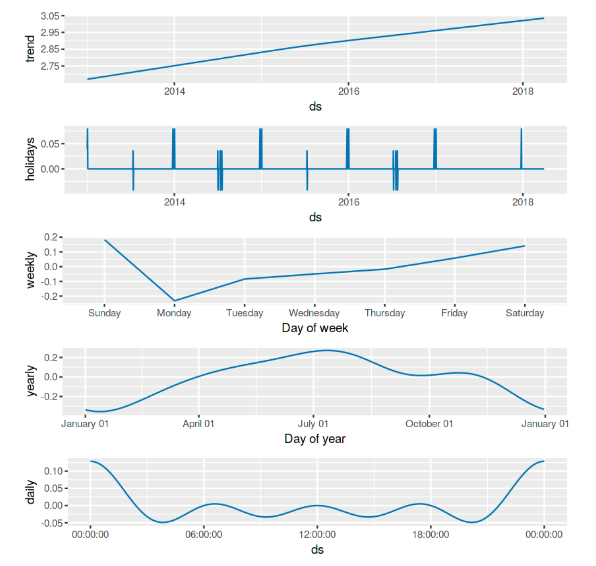
**Interpreting the Results**

We see the model has improved a lot after optimising the prophet parameters and including holidays, seasonality and additional regressors in the data. The trend is NOT much fluctuating like the baseline models and there is **NO CHANGE POINTS** of sales as well after fitting a better model. The model is NOT much over fitting as well. We can conclude that Holidays has an effect on Sales and we have taken care of it in our optimised models. We can see there is a Drop of Sales from Sunday to Monday. Therefore there must be holiday effect on our sales data. There is peak in sales in July that means those may the festive times or seasonal sales with high discount prices.

**Results without Holiday and Seasonality**



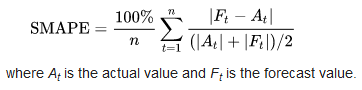
**Results with Holiday and Seasonality**



**Model comparison**

1. **Symmetric Mean Absolute Percent Error (SMAPE)** is an alternative to Mean Absolute Percent Error (MAPE) when there are zero or near-zero demand for items.

2. SMAPE self-limits to an error rate of 200%, reducing the influence of these low volume items.

****

We used **Prophet** Model and **ARIMA** model for predicting the sales values and used **SMAPE** to judge the accuracy of these models.

Prophet model: SMAPE= **20.2%** AND ARIMA model: SMAPE=**25.5%**

So **Prophet Model** had better accuracy, the reason could be inclusion of holidays and better prediction of seasonality with prophet model. Holidays and seasonality explained the shocks in sales values therefore further optimisation on these factors can improve the accuracy of the model.

**Recommendations**

1. More optimising the parameters using **Bayesian Optimisation**.

2. Include Multiplicative Seasonality in the model.

3. Remove the outliers and adjust Trend Flexibility.

4. By default Prophet uses **LINEAR GROWTH** for training the model. But we can use the **LOGISTIC GROWTH** in case of Multiplicative Seasonality