Forecasting Rossman Store Sales

From Kaggle Competitions

Amit Behura

Introduction to the dataset:

The dataset has been taken from kaggle competitions. Dataset contains 27 months daily sales information from 1115 stores including many factors like promotions, competition, school and state holidays.

Note: Only Train dataset is considered in this project. Store dataset includes locality and seasonality factor, which can improve the solutions.

Dependent variable required is defined by “Sales” which provides daily sales. Other dependent variable is “Customers”, since it can’t be predicted for future and Sales is closely dependent on this variable.

Structure and summary of the dataset:

```
> str(Train)
Classes 'data.table' and 'data.frame':  1017209 obs. of  9 variables:
 $ Store         : int  1 2 3 4 5 6 7 8 9 10 ...
 $ DayOfWeek     : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Date          : IDate, format: "2015-07-31" "2015-07-31" "2015-07-31" ...
 $ Sales         : int  5263 6064 8314 13995 4822 5651 15344 8492 8565 7185 ...
 $ Customers     : int  555 625 821 1498 559 589 1414 833 687 681 ...
 $ Open          : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Promo         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ StateHoliday : Factor w/ 4 levels "0","a","b","c": 1 1 1 1 1 1 1 1 1 1 ...
 $ SchoolHoliday: int  1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, ".internal.selfref")=<externalptr>
> summary(Train)
     Store          DayOfWeek          Date                Sales          Customers
 Min.   :   1.0   Min.   :1.000   Min.   :2013-01-01   Min.   :    0   Min.   :   0.0
 1st Qu.: 280.0   1st Qu.:2.000   1st Qu.:2013-08-17   1st Qu.: 3727   1st Qu.: 405.0
 Median : 558.0   Median :4.000   Median :2014-04-02   Median : 5744   Median : 609.0
 Mean   : 558.4   Mean   :3.998   Mean   :2014-04-11   Mean   : 5774   Mean   : 633.1
 3rd Qu.: 838.0   3rd Qu.:6.000   3rd Qu.:2014-12-12   3rd Qu.: 7856   3rd Qu.: 837.0
 Max.   :1115.0   Max.   :7.000   Max.   :2015-07-31   Max.   :41551   Max.   :7388.0
      Open             Promo        StateHoliday SchoolHoliday
 Min.   :0.0000   Min.   :0.0000   0:986159     Min.   :0.0000
 1st Qu.:1.0000   1st Qu.:0.0000   a: 20260     1st Qu.:0.0000
 Median :1.0000   Median :0.0000   b:  6690     Median :0.0000
 Mean   :0.8301   Mean   :0.3815   c:  4100     Mean   :0.1786
 3rd Qu.:1.0000   3rd Qu.:1.0000                3rd Qu.:0.0000
 Max.   :1.0000   Max.   :1.0000                Max.   :1.0000
```

Date variable is not in proper date format due to direct import to data frame. Also there are no NAs in the dataset so far. Further due to computational power limitations, analysis would be done using very tiny subset of whole information.

Pre-processing:

```
> subsetx <- filter(Train, Store<5)
> str(subsetx)
Classes 'data.table' and 'data.frame':  3768 obs. of  9 variables:
 $ Store        : int  1 2 3 4 1 2 3 4 1 2 ...
 $ DayOfWeek    : int  5 5 5 5 4 4 4 4 3 3 ...
 $ Date         : IDate, format: "2015-07-31" "2015-07-31" "2015-07-31" ...
 $ Sales        : int  5263 6064 8314 13995 5020 5567 8977 10387 4782 6402 ...
 $ Customers    : int  555 625 821 1498 546 601 823 1276 523 727 ...
 $ Open         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Promo        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ StateHoliday : Factor w/ 4 levels "0","a","b","c": 1 1 1 1 1 1 1 1 1 1 ...
 $ SchoolHoliday: int  1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

```
> summary(subsetx)
     Store          DayOfWeek          Date                Sales          Customers
 Min.   :1.00   Min.   :1.000   Min.   :2013-01-01   Min.   :    0   Min.   :   0.0
 1st Qu.:1.75   1st Qu.:2.000   1st Qu.:2013-08-24   1st Qu.: 3724   1st Qu.: 468.0
 Median :2.50   Median :4.000   Median :2014-04-16   Median : 5256   Median : 629.0
 Mean   :2.50   Mean   :3.998   Mean   :2014-04-16   Mean   : 5458   Mean   : 668.5
 3rd Qu.:3.25   3rd Qu.:6.000   3rd Qu.:2014-12-08   3rd Qu.: 7901   3rd Qu.: 901.0
 Max.   :4.00   Max.   :7.000   Max.   :2015-07-31   Max.   :17412   Max.   :2216.0
      Open            Promo        StateHoliday SchoolHoliday
 Min.   :0.0000   Min.   :0.0000   0:3663       Min.   :0.0000
 1st Qu.:1.0000   1st Qu.:0.0000   a:  65       1st Qu.:0.0000
 Median :1.0000   Median :0.0000   b:  24       Median :0.0000
 Mean   :0.8301   Mean   :0.3822   c:  16       Mean   :0.1866
 3rd Qu.:1.0000   3rd Qu.:1.0000                3rd Qu.:0.0000
 Max.   :1.0000   Max.   :1.0000                Max.   :1.0000
```

After filtering dataset for 4 store, we have 3768 observations with 9 variables. Further there is no NAs available in the data, so we don't need to mutate NAs for further analysis.

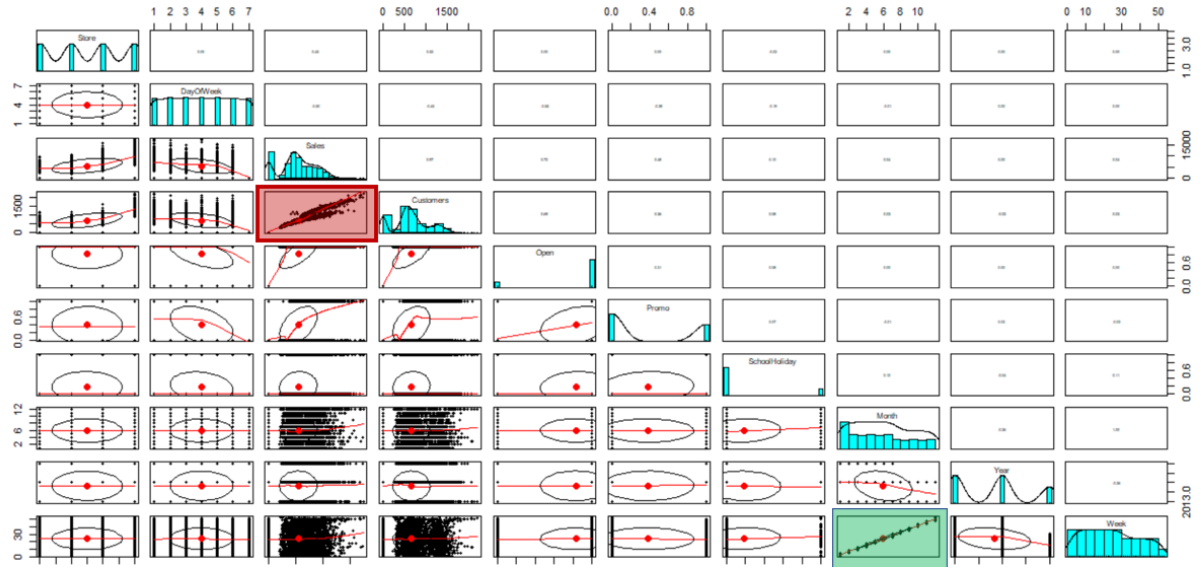
```
> subsetx$Date <- as.Date(subsetx$Date)
> subsetx$Month <- month(subsetx$Date)
> subsetx$Year <- year(subsetx$Date)
> subsetx$Week <- week(subsetx$Date)
> summary(subsetx)
     Store          DayOfWeek          Date                Sales          Customers
 Min.   :1.00   Min.   :1.000   Min.   :2013-01-01   Min.   :    0   Min.   :   0.0
 1st Qu.:1.75   1st Qu.:2.000   1st Qu.:2013-08-24   1st Qu.: 3724   1st Qu.: 468.0
 Median :2.50   Median :4.000   Median :2014-04-16   Median : 5256   Median : 629.0
 Mean   :2.50   Mean   :3.998   Mean   :2014-04-16   Mean   : 5458   Mean   : 668.5
 3rd Qu.:3.25   3rd Qu.:6.000   3rd Qu.:2014-12-08   3rd Qu.: 7901   3rd Qu.: 901.0
 Max.   :4.00   Max.   :7.000   Max.   :2015-07-31   Max.   :17412   Max.   :2216.0
      Open            Promo        StateHoliday SchoolHoliday       Month
 Min.   :0.0000   Min.   :0.0000   0:3663       Min.   :0.0000   Min.   : 1.000
 1st Qu.:1.0000   1st Qu.:0.0000   a:  65       1st Qu.:0.0000   1st Qu.: 3.000
 Median :1.0000   Median :0.0000   b:  24       Median :0.0000   Median : 6.000
 Mean   :0.8301   Mean   :0.3822   c:  16       Mean   :0.1866   Mean   : 5.962
 3rd Qu.:1.0000   3rd Qu.:1.0000                3rd Qu.:0.0000   3rd Qu.: 9.000
 Max.   :1.0000   Max.   :1.0000                Max.   :1.0000   Max.   :12.000
      Year          Week
 Min.   :2013   Min.   : 1.00
 1st Qu.:2013   1st Qu.:12.00
 Median :2014   Median :23.00
 Mean   :2014   Mean   :24.11
 3rd Qu.:2014   3rd Qu.:36.00
 Max.   :2015   Max.   :53.00
```

From date variable, Date-month-week of the year factors need to be extracted to include them as independent variables independently. Then we divide the data into train and test by taking first 25 months to be former and last 6 months to be later. (8:2 ratio)

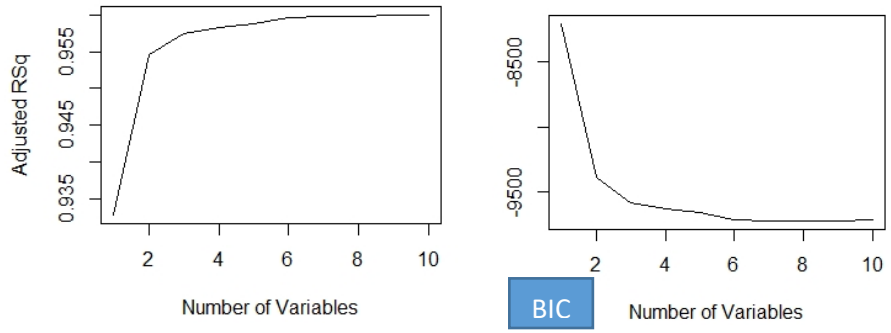
Correlation Matrix:

Before proceeding for advanced modelling, we need to extract efficient features to explain variance of Sales. Only taking variable with numerical information, higher

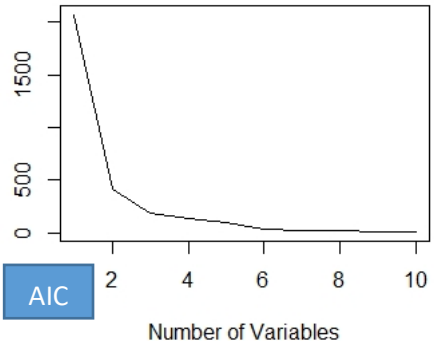
correlation between Sales & customers, and Week & Months is observed. So we need to analyse further to decide on which variables to be excluded.



Variable Power explained by Adjusted R-Square, BIC & AIC:



BIC

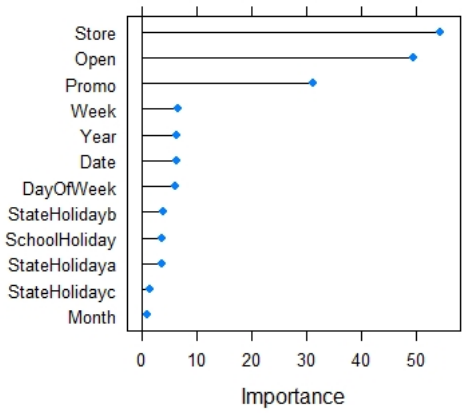


AIC

While taking max number of variables to be 10. The most efficient number of variables needed by measure of Adjusted R-Square and AIC is 10. For BIS, it is 9. After checking feature importance, we can finalise the list of independent variables need to used.

```
control <- trainControl(method="repeatedcv", number=5)
model <- train(Sales~., data=train, method="lm", preProcess="scale", trControl=control)
importance <- varImp(model, scale=FALSE)
importance
plot(importance)
```

```
> importance <- varImp(model, scale=FALSE)
> print(importance)
lm variable importance

             Overall
Store         54.410
Open          49.479
Promo         31.162
Week           6.665
Year           6.388
Date           6.386
DayOfWeek      6.023
StateHolidayb  3.864
SchoolHoliday  3.827
StateHolidaya  3.735
StateHolidayc  1.427
Month          1.016
```



- Customer Variable is excluded due to dependent variable nature which can't be taken for future values.
- Month is excluded due to perfect correlation with Week and having very low importance explained using linear relations.

Regression Modelling:

```
> NewModel <- train(Sales~Open+Promo+DayOfWeek+Week+Date+Year+Store, data=train,
+                   method="lm", preProcess="scale", trControl=control)
> predictions <- predict(NewModel, test)
> accuracy(predictions,test$Sales)
                ME    RMSE      MAE MPE MAPE
Test set 123.7228 1463.87 1099.498 NaN  Inf
```

Linear Regression Model

```
> NewModel2 <- train(Sales~Open+Promo+DayOfWeek+Week+Date+Year+Store, data=train,
+                    method="penalized", preProcess="scale", trControl=control)
> predictions2 <- predict(NewModel2, test)
> accuracy(predictions2,test$Sales)
                ME     RMSE      MAE MPE MAPE
Test set 124.3421 1461.546 1089.053 NaN  Inf
```

Penalized Regression Model

```
> NewModel3 <- train(Sales~Open+Promo+DayOfWeek+Week+Date+Year+Store, data=train,
+                    method="lasso", preProcess="scale", trControl=control)
> predictions3 <- predict(NewModel3, test)
> accuracy(predictions3,test$Sales)
                ME     RMSE      MAE MPE MAPE
Test set 123.8231 1462.215 1096.814 NaN  Inf
```

Lasso Regression Model

Three regression models are trained using cross validation method with 5 iterations. By RMSE, Penalized Regression Model is providing most efficient model.

Note:

- RMSE have 4 digit value due the range of sales, which is in 5 digit range.
- MAPE is infinite, due to one limitation of "MAPE". In case of large number of 0 values in the actual data, MAPE shows infinite due to large difference with value predicted by regression models.

	ME	RMSE	MAE	MPE	MAPE
method="lm"	123.7228	1463.87	1099.498	NaN	Inf
method="penalized"	124.3421	1461.546	1089.053	NaN	Inf
method="lasso"	123.8231	1462.215	1096.814	NaN	Inf

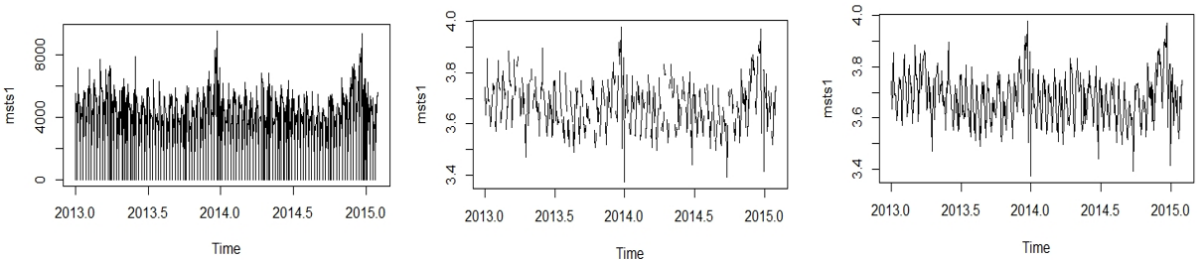
Time Series Analysis

```
TS1 <- rev(filter(train,Store==1)$Sales)
Test1 <- rev(filter(test,Store==1)$Sales)

msts1 <- msts(TS1,seasonal.periods = c(7,365.25),start = decimal_date(as.Date("2013-01-01")))
Test1 <- msts(Test1,seasonal.periods = c(7,365.25),start = decimal_date(as.Date("2015-02-01"))
msts1 <- log10(msts1)
msts1 <- do.call(data.frame,                      # Replace Inf in data by NA
                 lapply(msts1,
                        function(x) replace(x, is.infinite(x), NA)))
msts1 <-na.interp(msts1)
msts1 <- msts(msts1,seasonal.periods = c(7,365.25),start = decimal_date(as.Date("2013-01-01"))

ggseasonplot(msts1, year.labels=TRUE, year.labels.left=TRUE)
autoplot(stl(msts1,s.window = "periodic"))
```

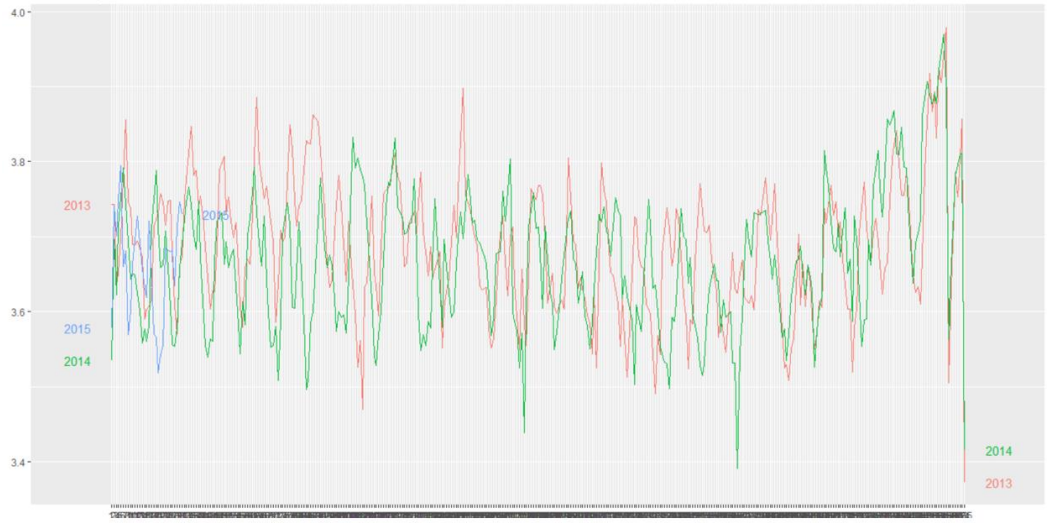
Training set: data from 1/1/2013 to 31/1/2015

Test Set: data from 1/2/2015 to 31/7/2015

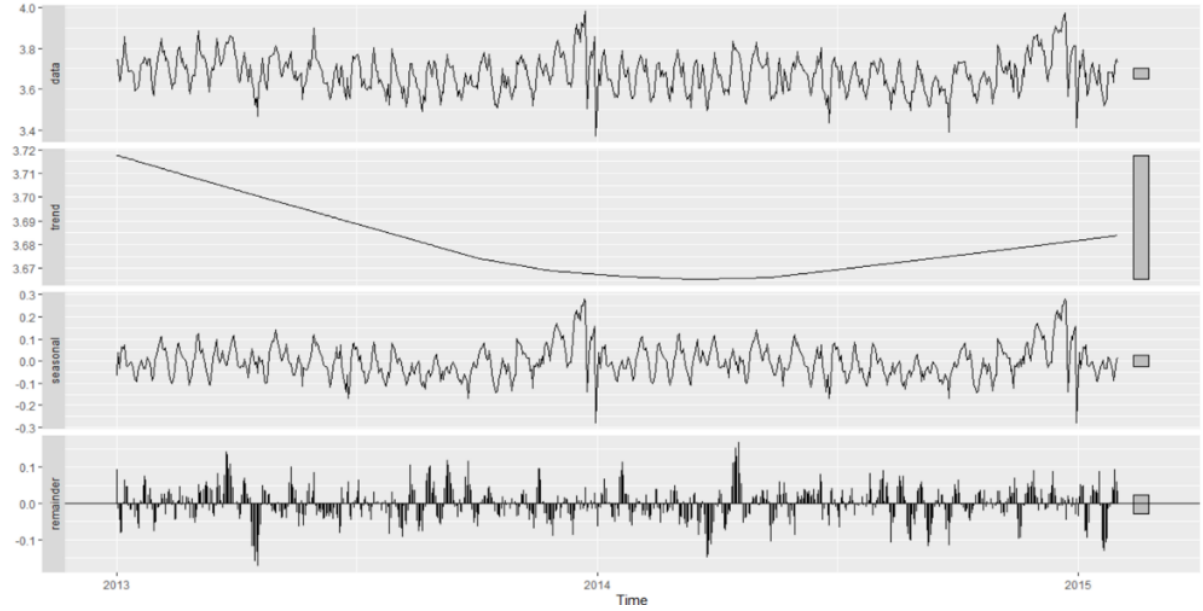
Time series model seems discontinuous due to huge number of zeroes for sales variable. To adjust the time-series, Log Transformation is done followed by interpolation for NAs from log transformations.



Seasonal and decomposition of time-series



No presence of year on year growth can be observed from above figure. As time series with respect to each year are crossing with each year, making there is no significant trend.



From above figure, we can conclude that there is no significant trend or cycle in the time-series. But in case of remainder part, the time-series is not a white noise. So we need to analyse the parts in more details to identify underlying patterns.

Benchmarking Time Series Forecasting

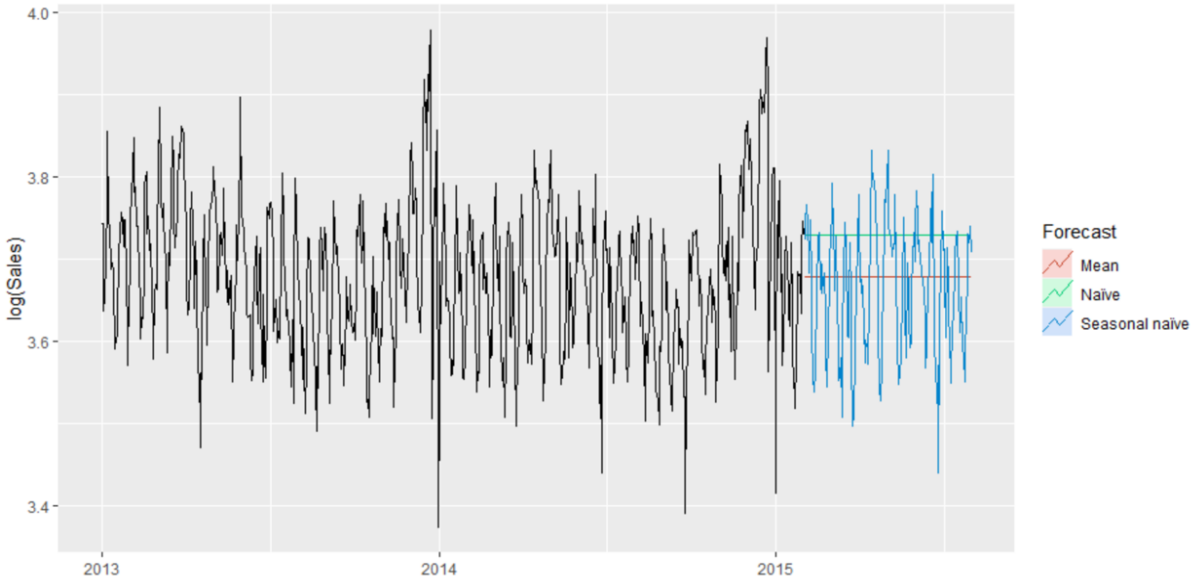
Setting a benchmark accuracy level is required when analysing sophisticated methods in terms of better performance identification. Here, three forecasting methods are used; namely Naïve, Mean Average and Naïve seasonal forecasting.

```
fit1 <- meanf(msts1,h=181)
fit2 <- rwf(msts1,h=181)
fit3 <- snaive(msts1,h=181)

autoplot(msts1) +
  autolayer(fit1, series="Mean", PI=FALSE) +
  autolayer(fit2, series="Naïve", PI=FALSE) +
  autolayer(fit3, series="Seasonal naïve", PI=FALSE) +
  xlab("Year") + ylab("log(Sales)") +
  ggtitle("Forecasts for Sales") +
  guides(colour=guide_legend(title="Forecast"))

predfit1 <- 10^(fit1$mean)
predfit2 <- 10^(fit2$mean)
predfit3 <- 10^(fit3$mean)
accuracy(predfit1,Test1)
accuracy(predfit2,Test1)
accuracy(predfit3,Test1)
```

With horizon of 181 (number of days in last 6 months), we predicted the sales value. Then reverse transformed it to the original range of sales value.

```
> accuracy(predfit1,Test1)
                   ME     RMSE      MAE  MPE MAPE       ACF1 Theil's U
Test set -1047.879 2151.456 1437.025 -Inf  Inf -0.1704529         0
> accuracy(predfit2,Test1)
                   ME     RMSE      MAE  MPE MAPE       ACF1 Theil's U
Test set -1646.403 2498.271 1789.939 -Inf  Inf -0.1704529         0
> accuracy(predfit3,Test1)
                   ME     RMSE      MAE  MPE MAPE     ACF1 Theil's U
Test set -971.1147 2146.868 1325.497 -Inf  Inf -0.15856         0
```

Seasonal naïve forecasting provides highest accuracy. So further models need to have RMSE less than 2146.868 to be considered for future forecasting.



Sophisticated Time-Series Forecasting

Three methods are considered here;

- Holt's linear smoothing
- Moving average (15 days)
- ARIMA

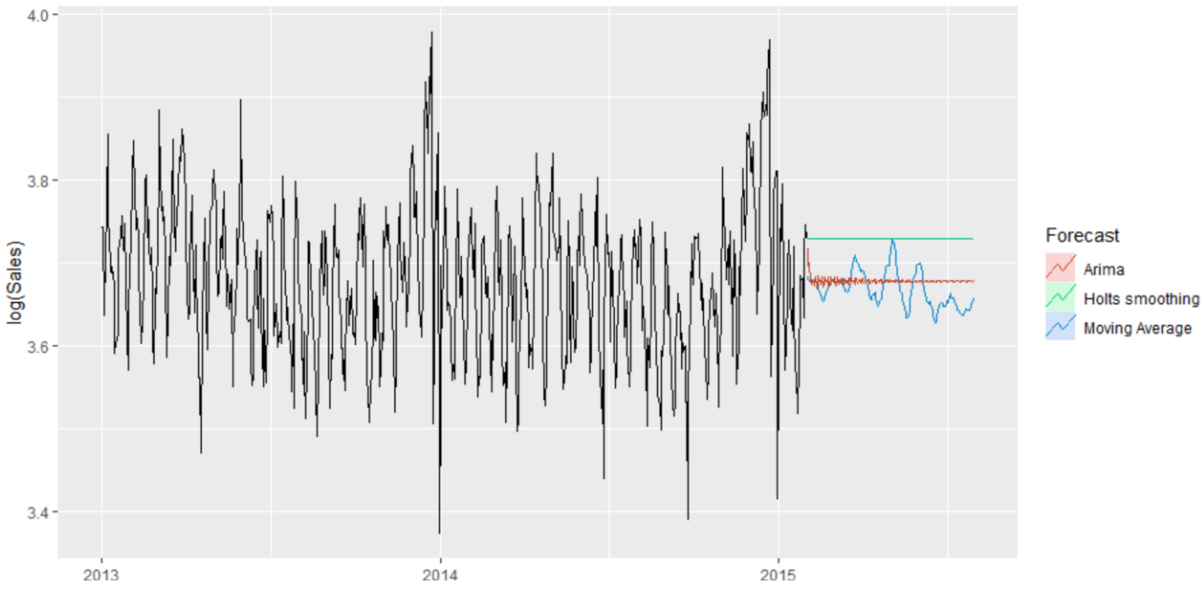
For ARIMA, benchmark model identified using auto.arima feature. Using the benchmark model only ARIMA is implemented.

```
> # Exponential Smoothing & ARIMA
> # Holt's Linear Smoothing, Moving Avergae last 15 days
> fit4 <- holt(msts1, damped = TRUE, h=181)
> fit5 <- forecast(ma(msts1,15),181)
> #Auto Arima with low AIC, AICc, BIC value
> auto.arima(msts1)
Series: msts1
ARIMA(4,0,3) with non-zero mean

Coefficients:
          ar1     ar2     ar3      ar4     ma1     ma2      ma3   mean
      -0.4670  0.8839  0.4281  -0.4538  1.3291  0.1644  -0.4110  3.678
s.e.   0.1628  0.1674  0.0678   0.1159  0.1724  0.3055   0.1643  0.007

sigma^2 estimated as 0.003191:  log likelihood=1110.38
AIC=-2202.76   AICc=-2202.52   BIC=-2161.05
```

```
fit6 <- forecast(Arima(msts1,order = c(4,0,3)),h=181)
autoplot(msts1) +
  autolayer(fit4, series="Holts smoothing", PI=FALSE) +
  autolayer(fit5, series="Moving Average", PI=FALSE) +
  autolayer(fit6, series="Arima", PI=FALSE) +
  xlab("Year") + ylab("log(Sales)") +
  ggtitle("Forecasts for Sales") +
  guides(colour=guide_legend(title="Forecast"))

autoplot(msts1) +
  autolayer(fit6, series="ARIMA", PI=FALSE)+
  guides(colour=guide_legend(title="Daily forecasts"))

predfit4 <- 10^(fit4$mean)
predfit5 <- 10^(fit5$mean)
predfit6 <- 10^(fit6$mean)
accuracy(predfit4,Test1)
accuracy(predfit5,Test1)
accuracy(predfit6,Test1)
```

```
> accuracy(predfit4,Test1)
                ME     RMSE      MAE  MPE MAPE       ACF1 Theil's U
Test set -1653.061 2502.664 1794.681 -Inf  Inf -0.1704477         0
> accuracy(predfit5,Test1)
                ME     RMSE      MAE  MPE MAPE       ACF1 Theil's U
Test set -947.0415 2111.674 1386.668 -Inf  Inf -0.1646926         0
> accuracy(predfit6,Test1)
                ME     RMSE      MAE  MPE MAPE       ACF1 Theil's U
Test set -1052.703 2156.267 1437.801 -Inf  Inf -0.1721315         0
```

Only moving average method can be considered due to lower RMSE value compared to benchmark point.



Results

			Forecasting Method	ME	RMSE	MAE	MPE	MAPE
	Regression Models		method="lm"	123.7228	1463.87	1099.498	NaN	Inf
			method="penalized"	124.3421	1461.546	1089.053	NaN	Inf
			method="lasso"	123.8231	1462.215	1096.814	NaN	Inf
1	Time Series Models	Fit 1	Mean Average	-1047.879	2151.456	1437.025	-Inf	Inf
2		Fit 2	Naïve	-1646.403	2498.271	1789.939	-Inf	Inf
3		Fit 3	Seasonal Naive	-971.1147	2146.868	1325.497	-Inf	Inf
4		Fit 4	Exponential Smoothing	-1653.061	2502.664	1794.681	-Inf	Inf
5		Fit 5	Moving Average	-947.0415	2111.674	1386.668	-Inf	Inf
6		Fit 6	Arima	-1052.703	2156.267	1437.801	-Inf	Inf

Overall regression models have significant accuracy level compared to time series forecasting methods. The independent variable are necessary to taken into account while forecasting for future daily sales. Comparing all the models, penalized regression has the highest accuracy.

```
> stepmodel <- train(Sales~Open+Promo+DayOfWeek+Week+Date+Year+Store, data=train,
+                    method="penalized", preProcess="scale", trControl=control)

> predictions4 <- predict(stepmodel, test)
> pred2 <- data.frame(predictions4,test$Open)
> pred2$predictions4[which(pred2$test.Open == 0)] <- 0
> x <- as.numeric(pred2$predictions4)
> accuracy(x,test$Sales)
             ME     RMSE      MAE       MPE     MAPE
Test set 113.859 1295.952 846.7195 -3.726119 16.90585
```

- For the stores we selected a variable "OPEN"
- The store is open -> OPEN = 1
- The store is closed-> OPEN = 0
- Whenever the store is closed, the sales value are made as ZERO.

Open variable defines store is open (1) or closed (0) on certain date. With very large zero sales value available, a modification on penalized regression model is done. When a store is closed, model would forecast sales to be zero. Otherwise it would follow the penalized regression predicted values.

With that condition, more accuracy is achieved with RMSE level of 1295.952. From all the models used in this project, use of modified penalised regression model is recommended to be used for forecasting daily sales at Rossman stores.

Conclusion

- The pharmaceutical supply chain is highly complex and the demand is highly uncertain.
- Demand is influenced by a numerous external variable. For e.g., patients' length of treatment and air quality and hygiene in the surroundings.
- In this study, of all the models, regression-based forecasting model provides the best fitting curve to the historical train data with lower error estimates and higher accurate demand estimation.

- Analysis in the project suggests weekly forecasts in place of daily forecasts because the frequency of forecasting will reduce from 365.25 to 52 which will help in accommodating a better model in R.
- Weekly forecasts will save the money, time and efforts and they also increase the forecasting accuracy by aggregation. Considering the shelf life of the medicines, Reorder points can be scaled up through aggregate weekly forecasts resulting in saving ordering costs.