# ROSSMANN SALES PREDICTION

## Computing For Data Sciences-

## Final Project


GROUP 09

Anurag Patel(15BM6JP07)
Sachin Kumar(15BM6JP38)
Sanjeev Kumar(15BM6JP41)
Vivekanand Chadchan(15BM6JP52)

# Content

Competition Details

Impact of Solution

First Look at Data

Exploratory Data Analysis

Model Formulation

Challenges and Learning

# Competition Details

# Forecast sale using store, promotion and competitor data

- To forecast the daily sale of individual 1115 Rossmann stores located across Germany, 6 weeks in advance.
- Historical data upto 2 year 7 month is provided(Jan 2013 to July 2015)

<u>Evaluation Criteria:</u>
Submissions are evaluated on the Root Mean Square Percentage Error(RMSPE). Lower the score better will be the prediction

$$\mathrm{RMSPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2},$$

where $y_i$ denotes the sales of a single store on a single day and $y_i$_hat denotes the corresponding prediction. Anyday and store with 0 sales is ignored in scoring.

# Impact of Solution

- Better management of staff schedules.
- Provide enough time to store managers to focus on customers and their teams.
- Increase efficiency of employees.

# First Look at Data

# Details of Data Set Provided

| SNo | Data Set | Variables | No of Variables | No of observations |
|---|---|---|---|---|
| 1. | Train | store, day of week, date, sales,customers, open, promo, state holiday, school holiday | 9 | 1017210 |
| 2. | Store | store, storetype, assortment, competition distance, competition open since month, promo2, promo2since week, promo2since year, promo interval | 10 | 1115 |
| 3. | Test | id, store, dayofweek, date, open, promo, state holiday, school holiday | 8 | 41089 |

# After merging the variables of ‘train’ and ‘store’ dataset

| S No | Variables | Measurement Scale | Possible Values |
|---|---|---|---|
| 1. | Store | Nominal | 1 to 1115 |
| 2. | Dayofweek | Nominal | 1,2,3,4,5,6,7 |
| 3. | Date | Interval | 1/1/2013 to 7/31/2015 |
| 4. | Sales | Ratio | 0 to 41551 |
| 5. | Customers | Ratio | 0-7338 |
| 6. | Open | Nominal | 0 (closed),1 (open) |
| 7. | Promo | Nominal | 0(No Promotion), 1 (Offering Promotion) |

# Contd...

| S No | Variables | Measurement Scale* | Range |
|---|---|---|---|
| 8. | State holiday | Nominal | a: Public Holiday<br>b: Easter Holiday<br>c: Christmas Holiday<br>0: None |
| 9. | School holiday | Nominal | 0(No),1(Yes) |
| 10. | Store type | Nominal | a,b,c,d (Store models) |
| 11. | Assortment | Nominal | a: Basic<br>b: Extra<br>c: Extended |
| 12. | Competition distance | Ratio | 20-75860 |
| 13. | Competition open since month | Interval | 1(Jan) to 12(Dec) |

# Contd...

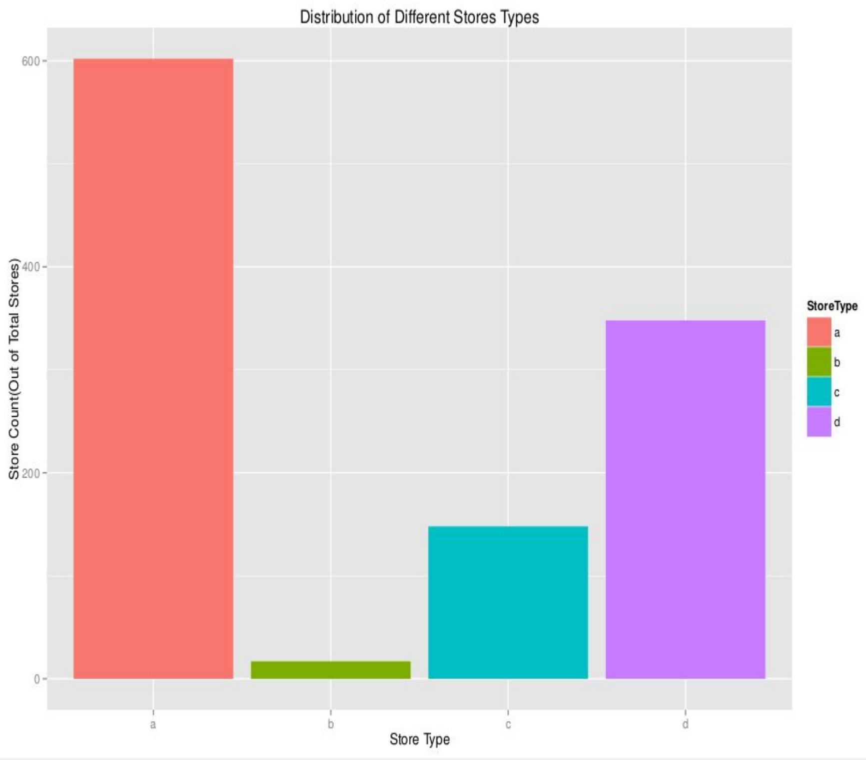
| | | | |
|---|---|---|---|
| 14. | Competition open since year | Interval | 1900-2015 |
| 15. | Promo2 (Long Term Promotion) | Nominal | 0,1 |
| 16. | Promo2 since week | Interval | 1-50 |
| 17. | Promo2 since year | Interval | 2009-2015 |
| 18. | Promo interval | Ordinal | (jan, apr, jul, oct)<br>(fab, may, aug, nov)<br>(mar, jun, sept, dec) |

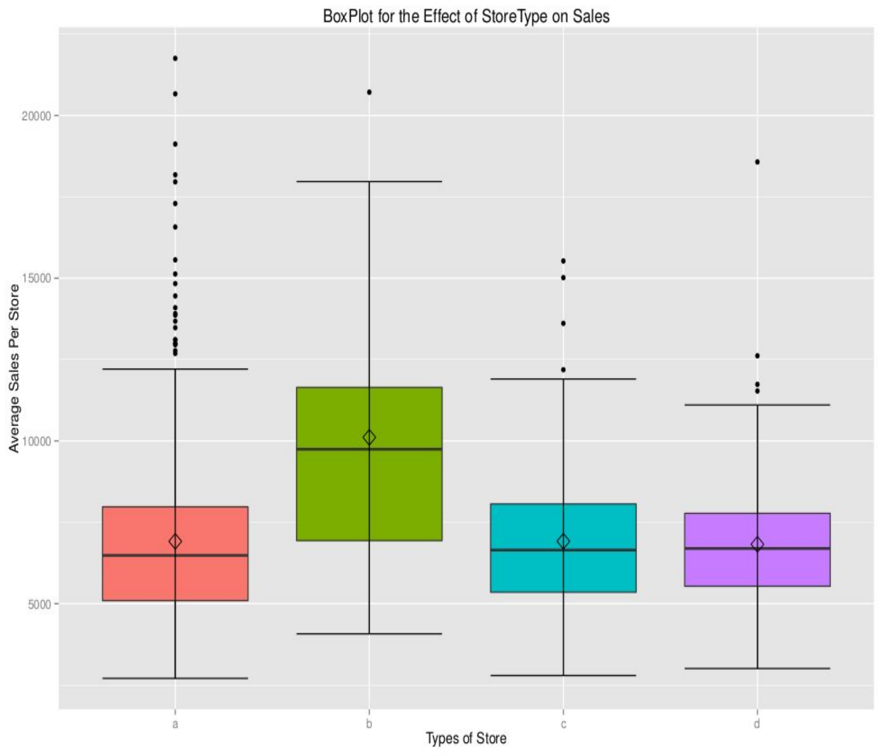
*Two Types of Variables in measurement scales

1. Categorical Variables : Nominal and Ordinal Scale
2. Numerical Variables : Interval and Ratio Scale

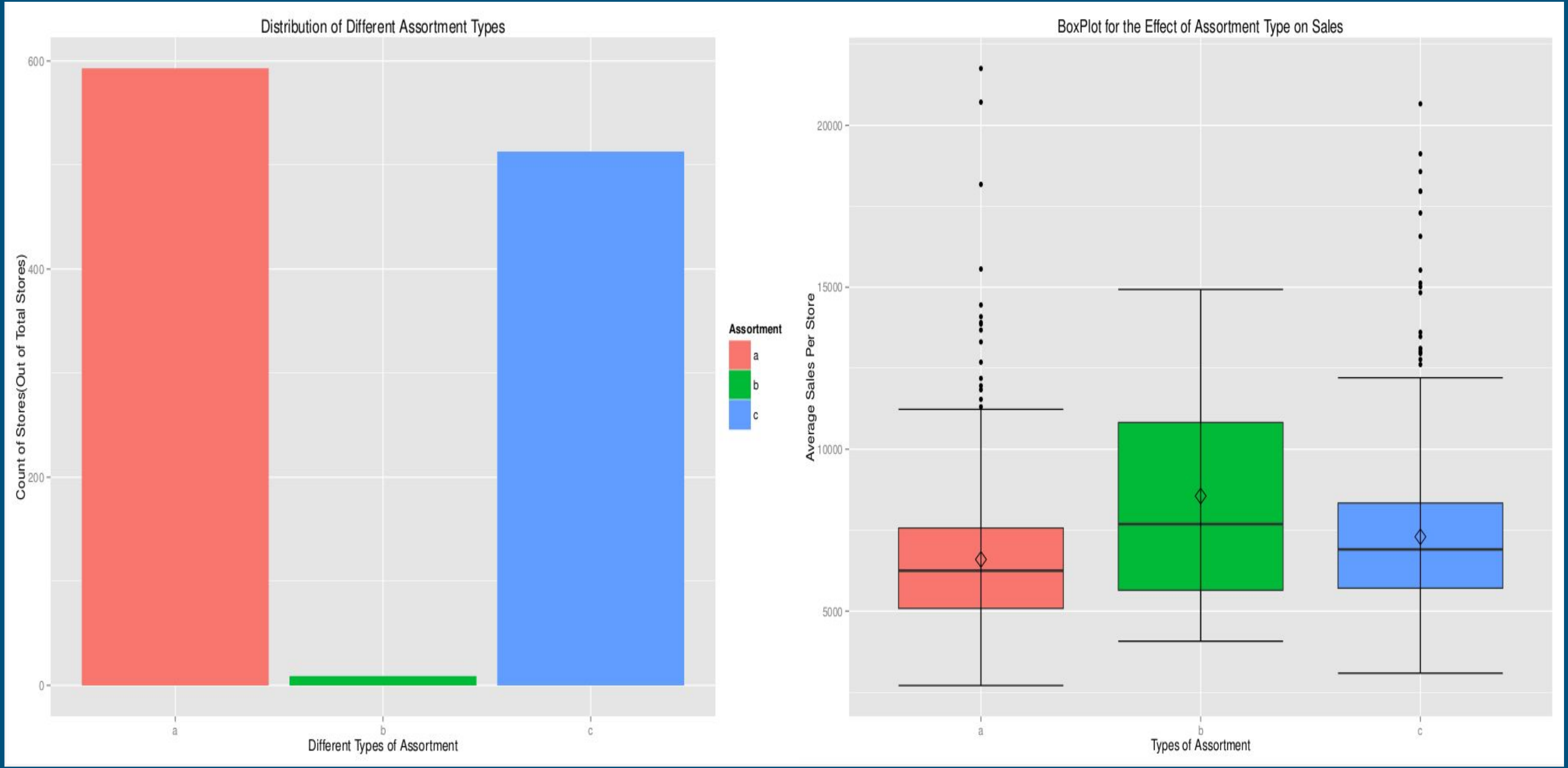
# Exploratory Data Analysis(EDA)

# Distribution of 4 store models and Average Sales



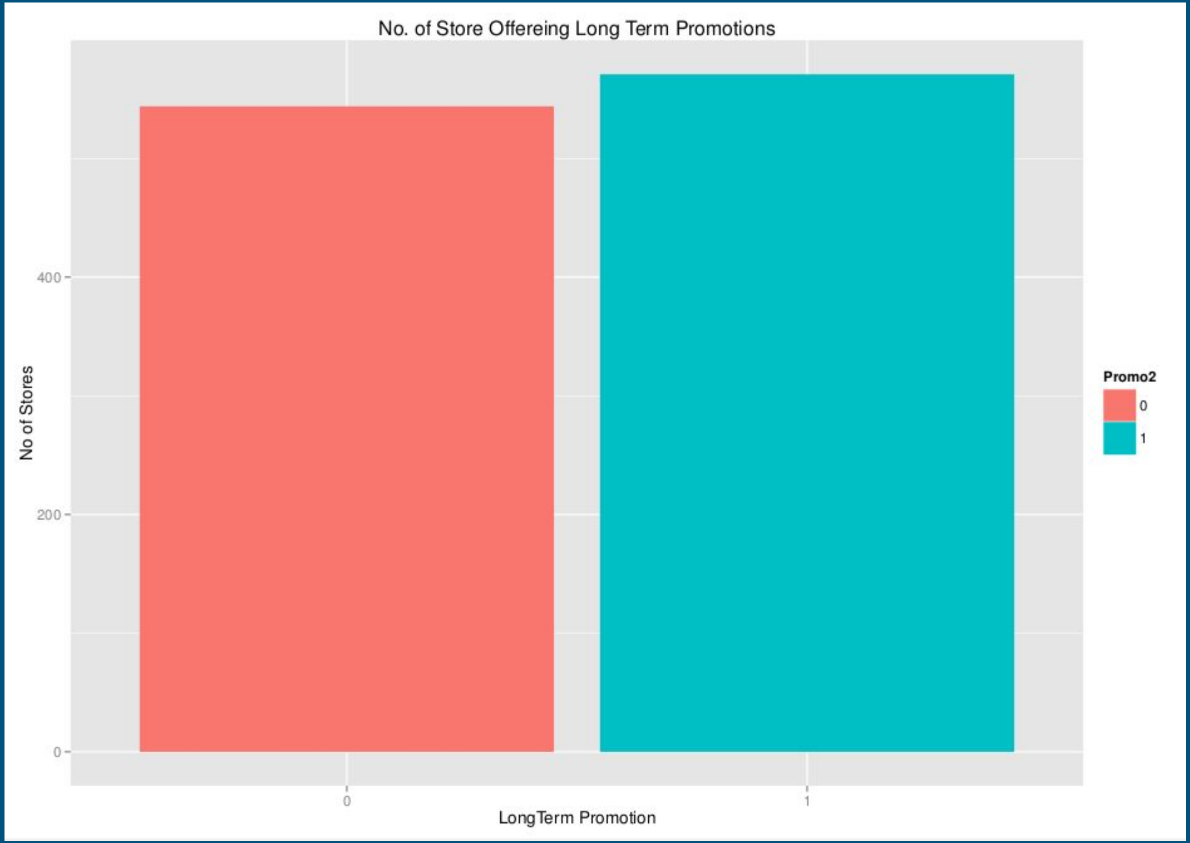
# Distribution of Assortment Type and Average Sales



# Type of Stores and Assortment Level(Contingency Table)

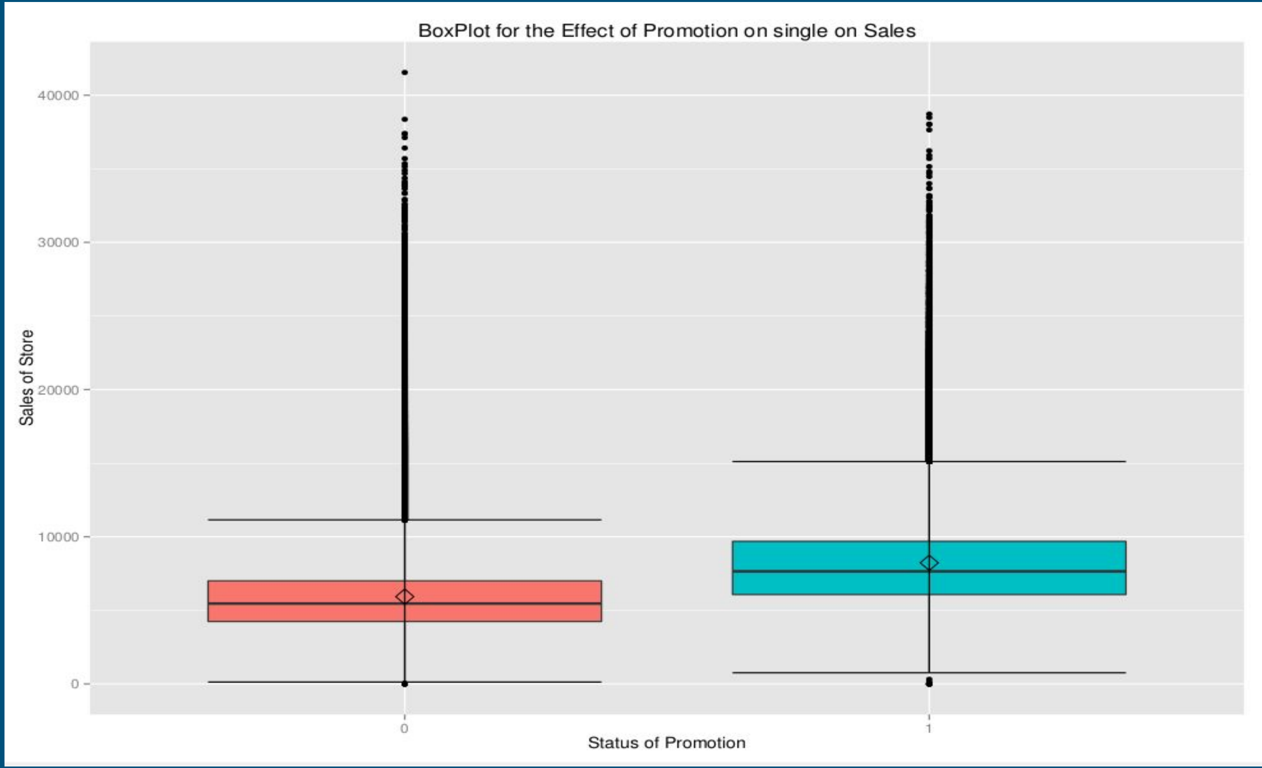
| StoreType\Assortment level | Assortment Level ‘a’ | Assortment Level ‘b’ | Assortment Level ‘c’ | **Total stores** |
|---|---|---|---|---|
| Store Type ‘a’ | 381 | 0 | 221 | **602** |
| Store Type ‘b’ | 7 | 9 | 1 | **17** |
| Store Type ‘c’ | 77 | 0 | 71 | **148** |
| Store Type ‘d’ | 128 | 0 | 220 | **348** |
| **Total Stores** | **593** | **9** | **513** | **1115** |

# Distribution of Stores involved in Long Term Promotion

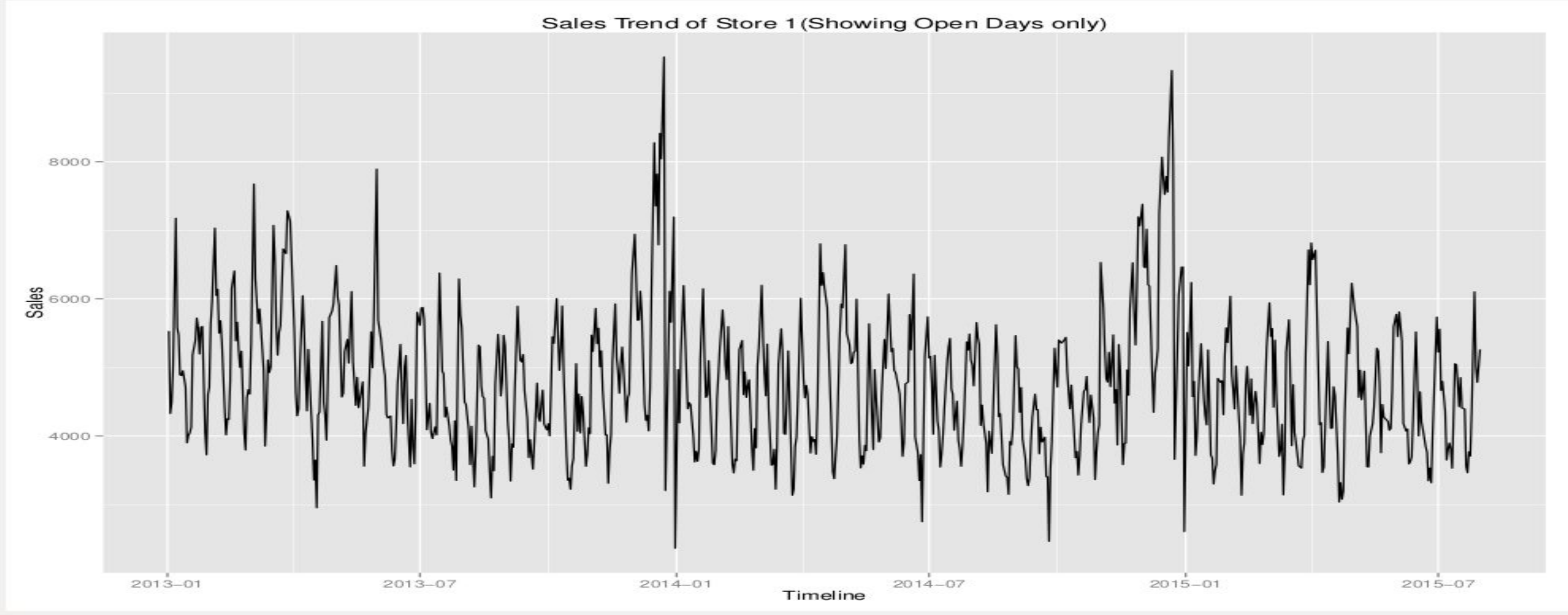



Yes : 571 stores
No : 544

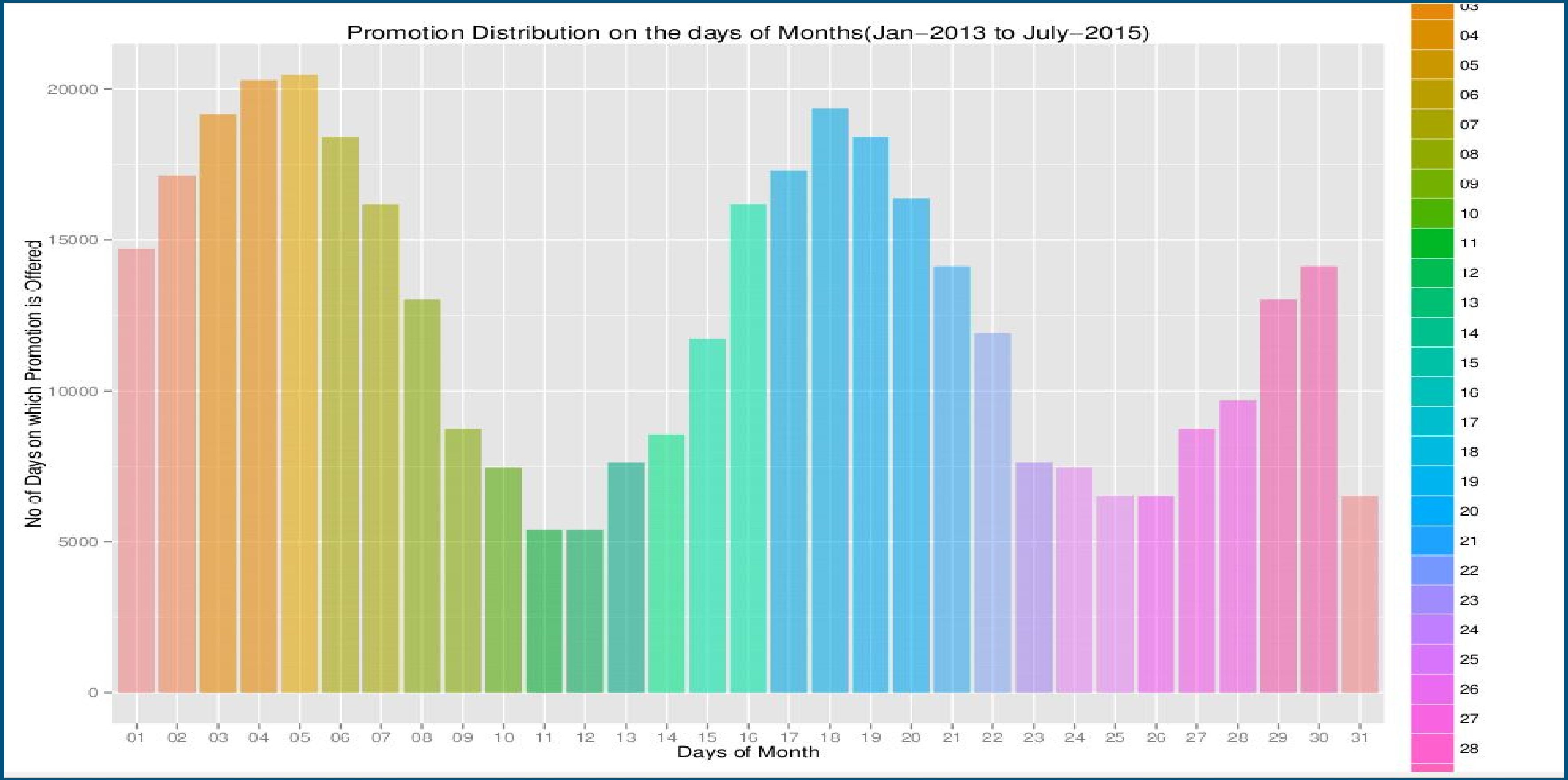
# Effect of single day Promotion on Sales



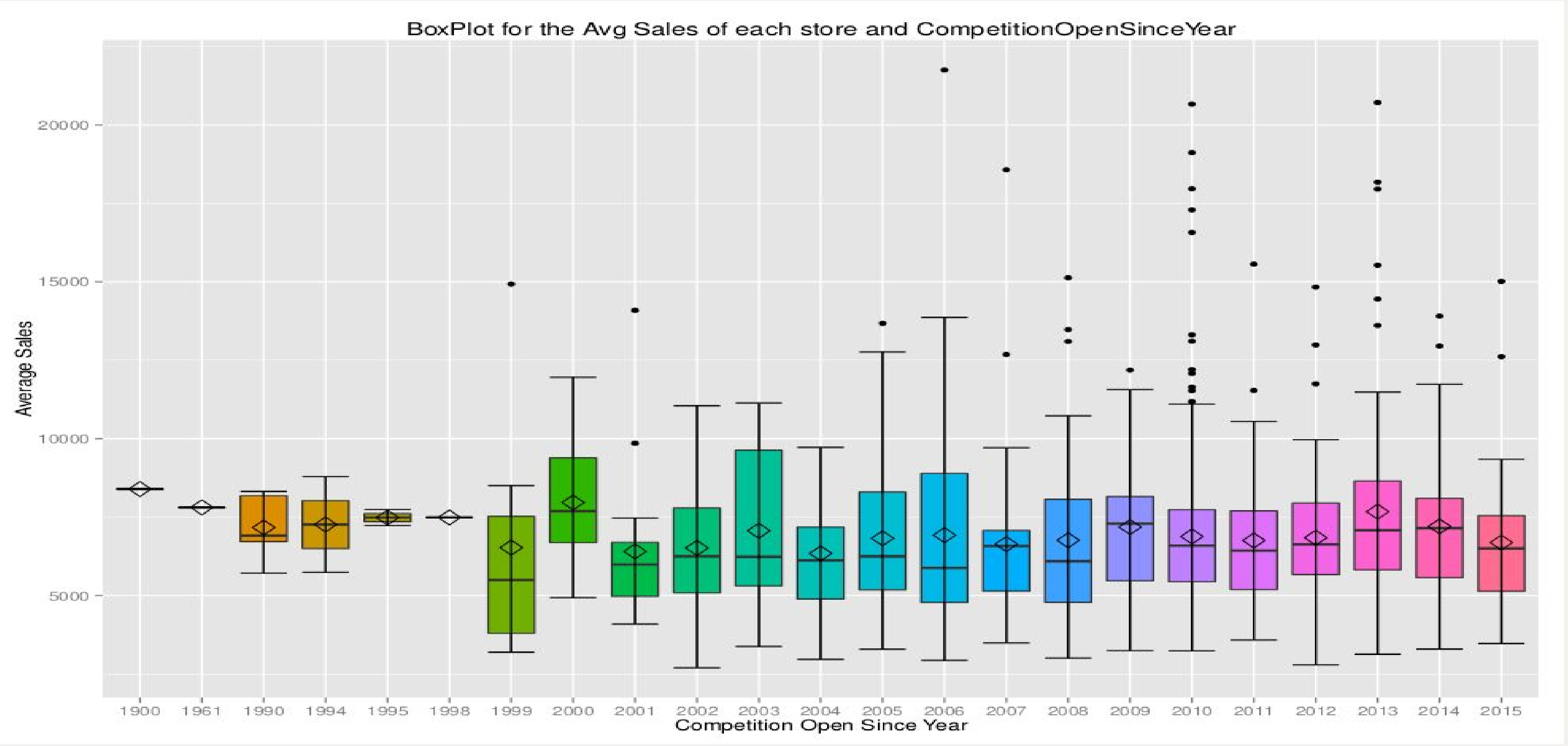
# Sales Trend of a Store(e.g Store 1)



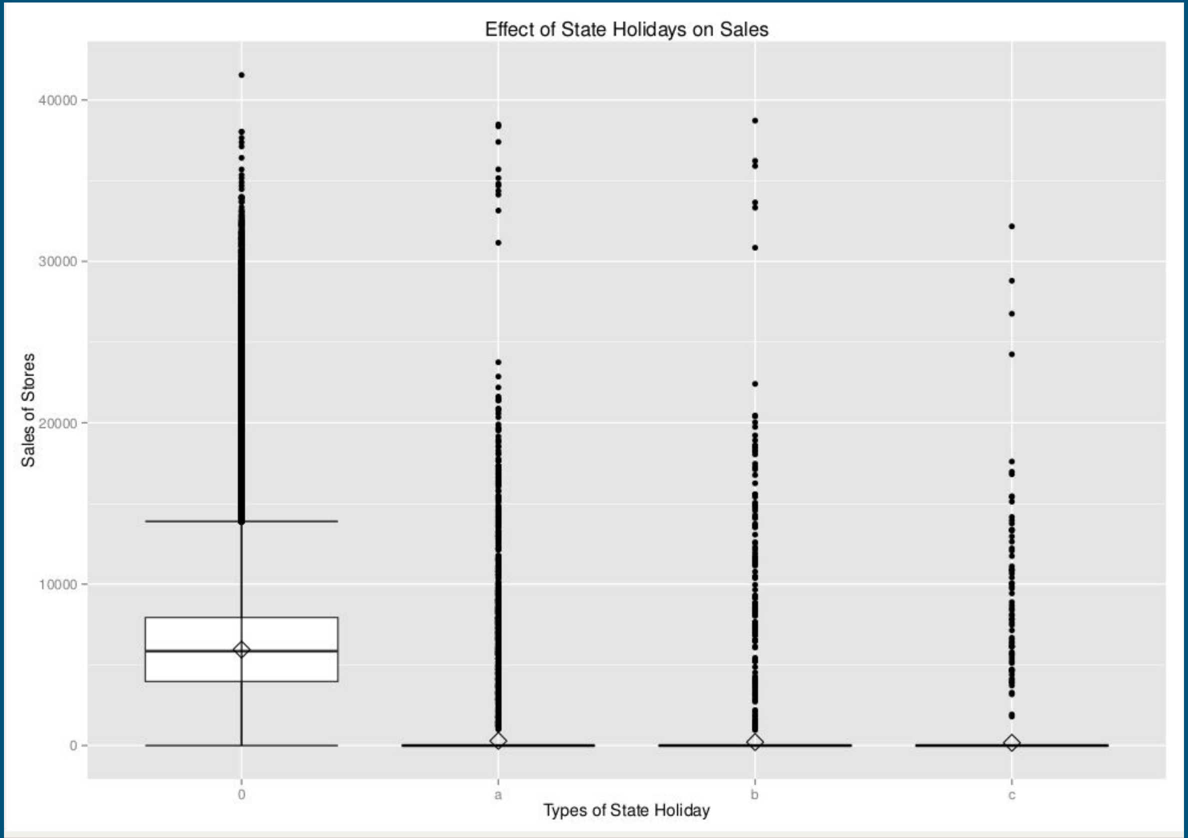
# Distribution of Promotion on each Day of Month



# Effect of Competition Open Since Year on Average Sales

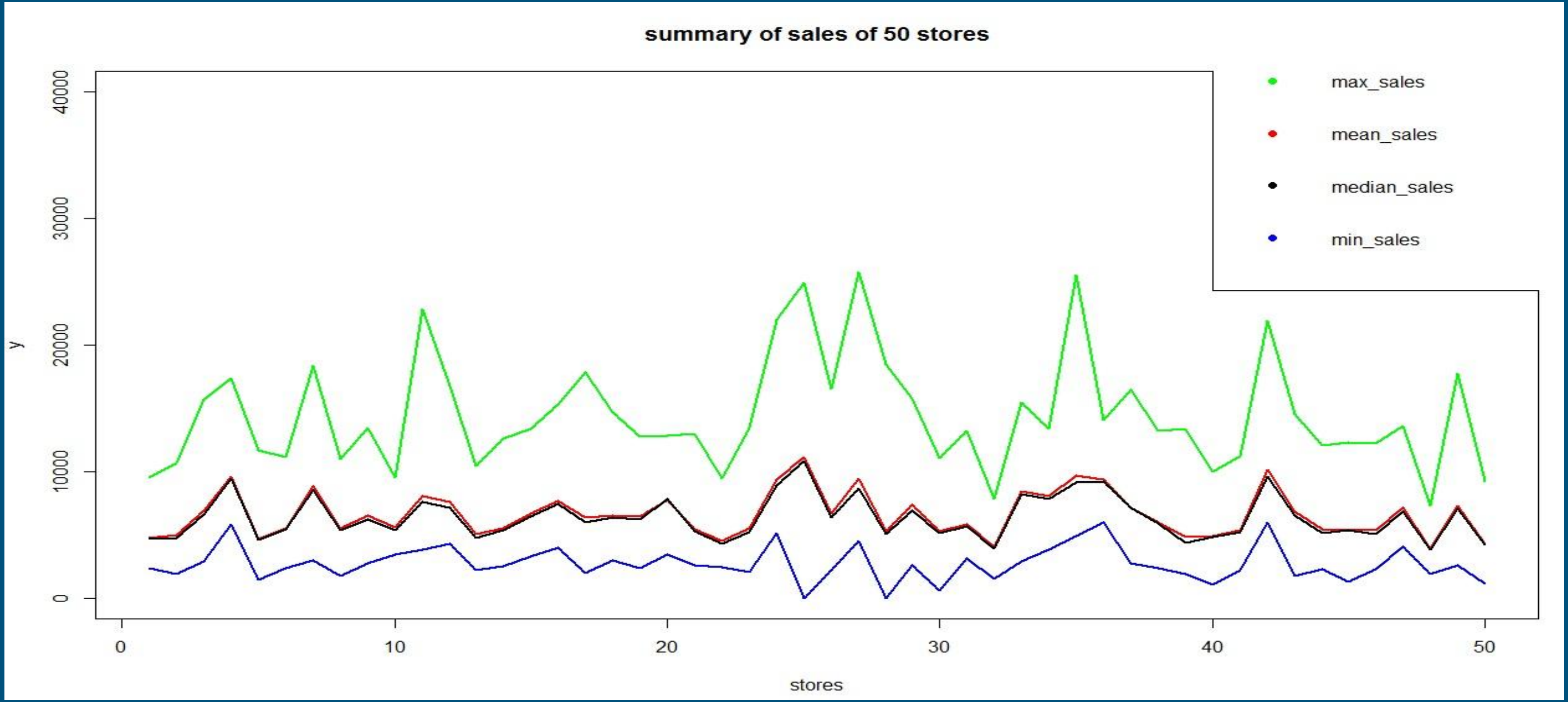


# Effect of State Holidays on Sales

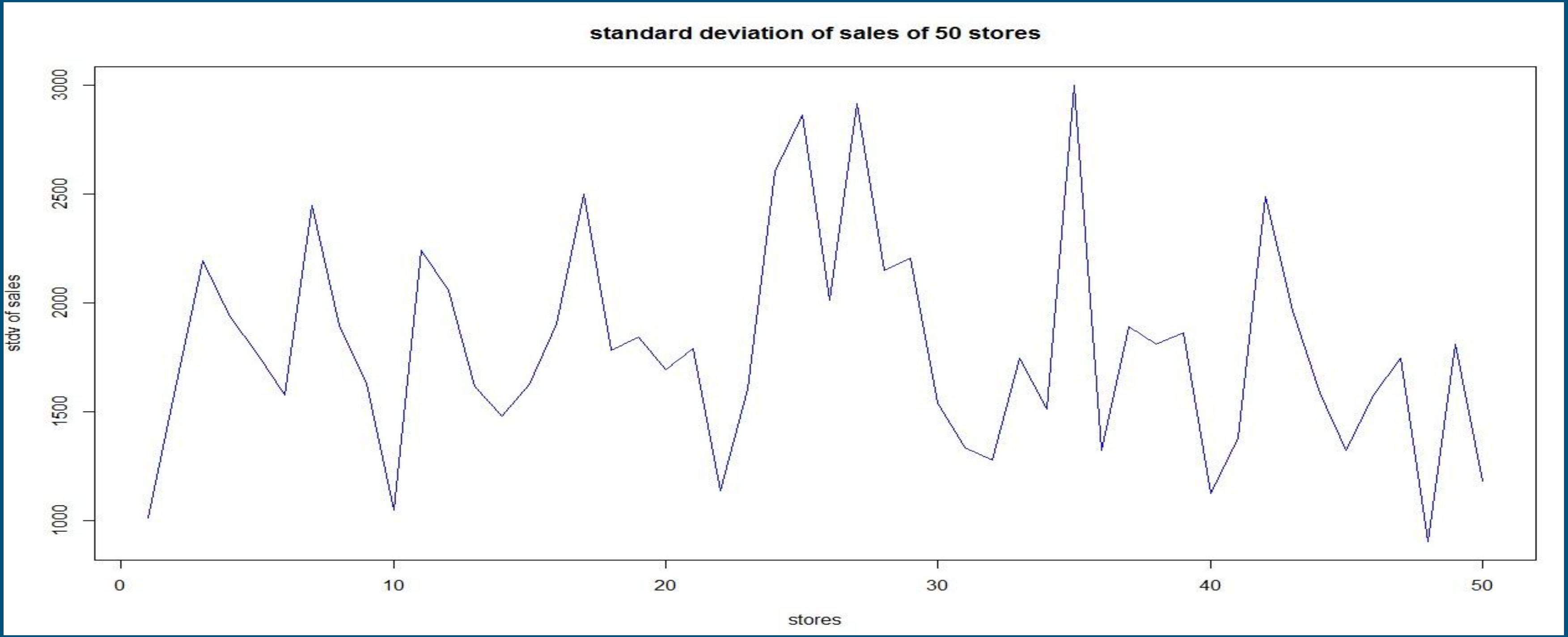



1. Majority of Stores are closed on state holidays.

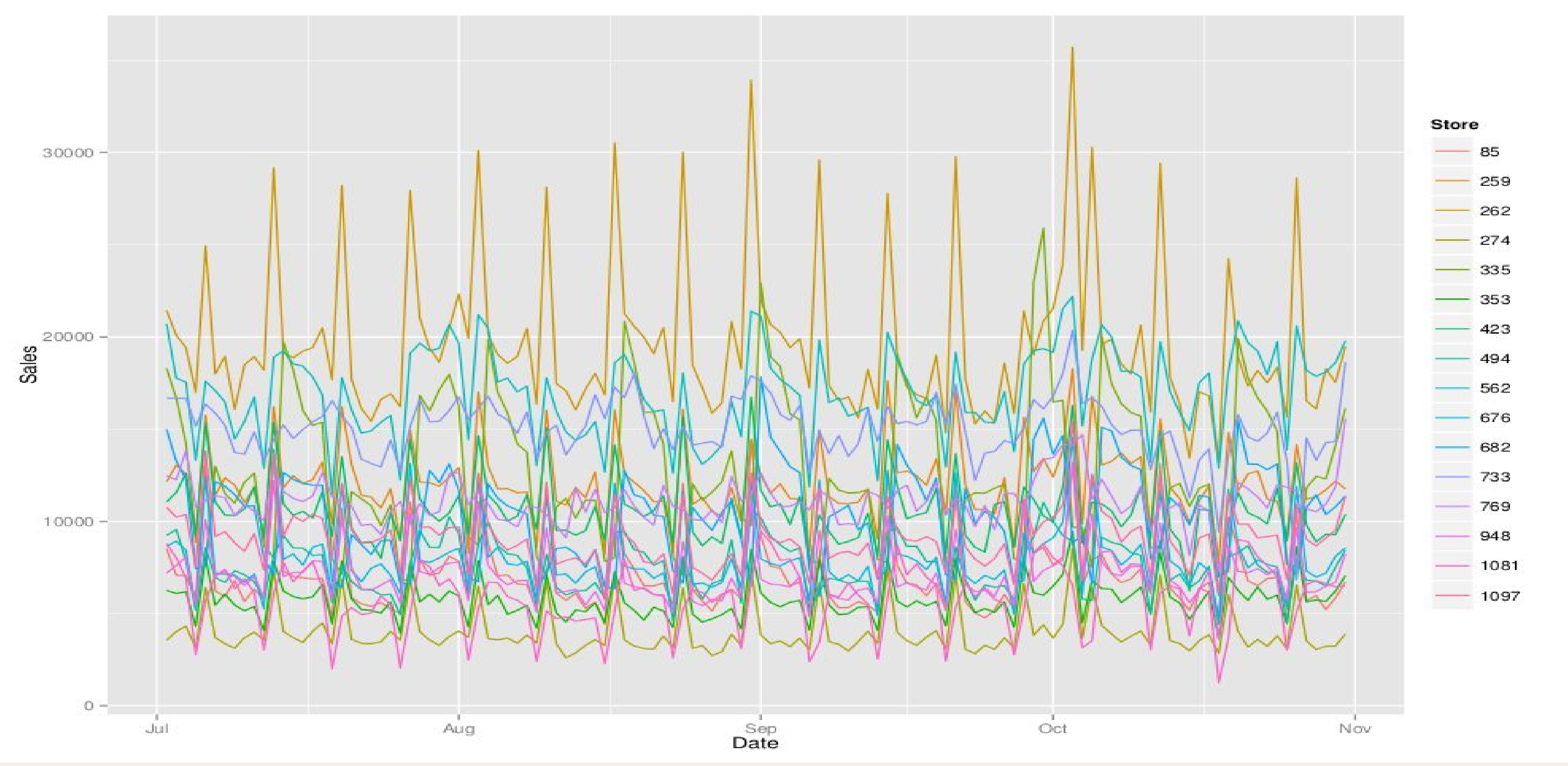
# Summary of Sales of 50 stores



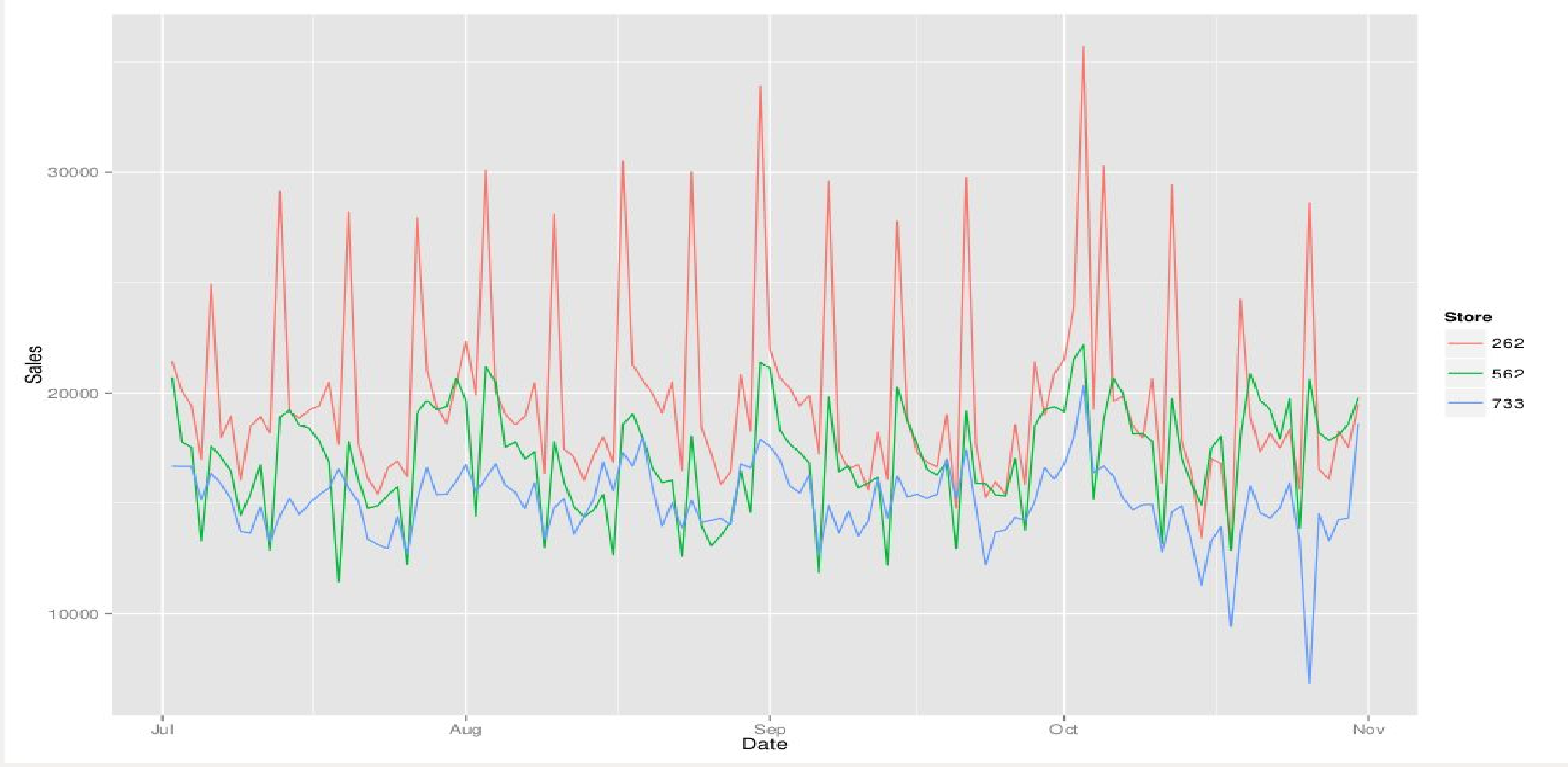
# Standard deviation of sales of 50 stores



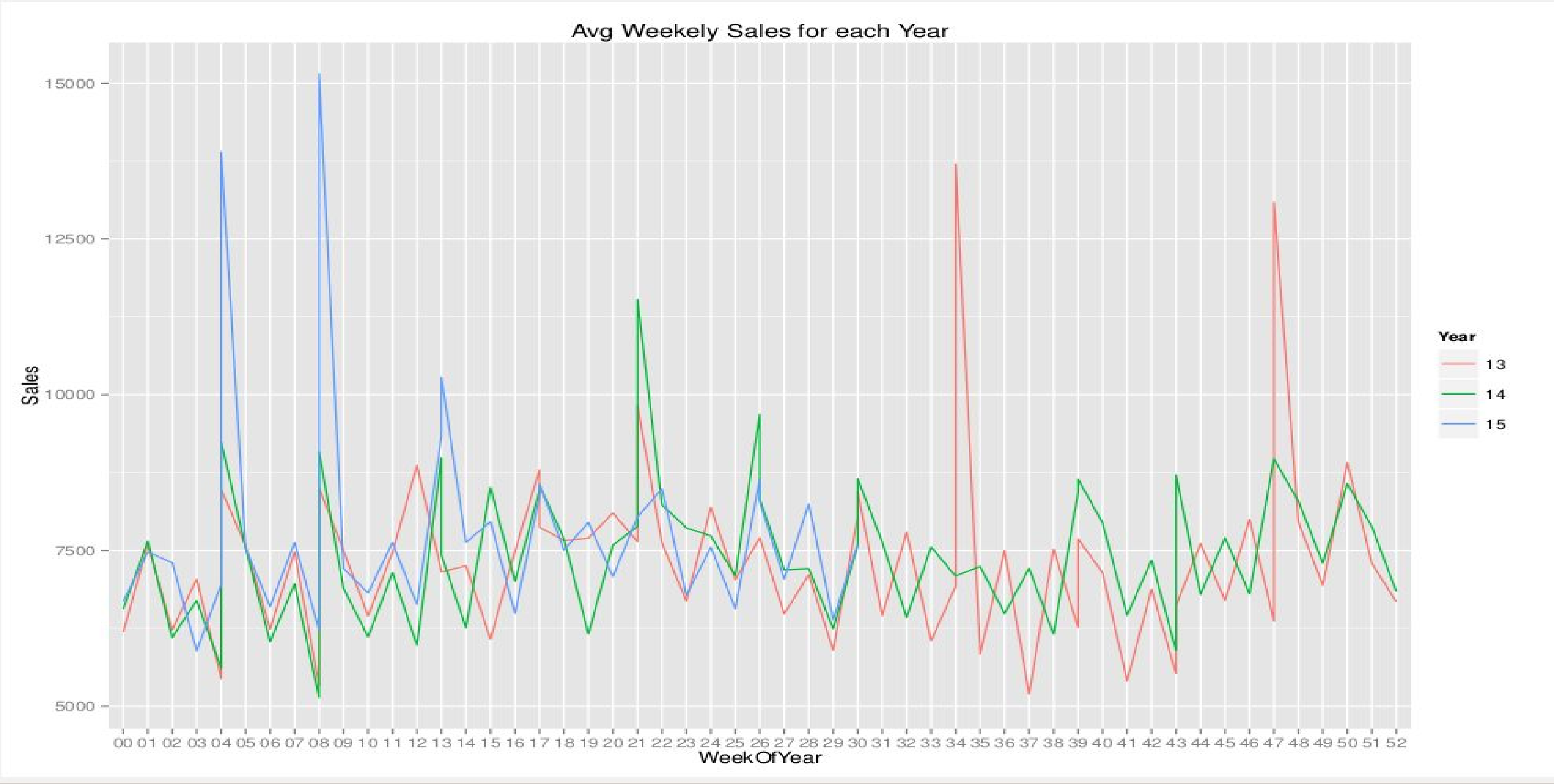
# Type B stores Sales data with time



# Further Zoom In



# Weekly Average Sales Distribution Store 85



# Inferences from EDA

- Type of Store plays an important role in opening pattern of stores. All Type ‘b’ stores never closed except for refurbishment or other reason.
- All Type ‘b’ stores have comparatively higher sales and it mostly constant with peaks appears on weekends.
- Assortment Level ‘b’ is only offered at Store Type ‘b’.
- Competition distance have -.55 (negative correlation) with average sales for stores offering assortment level ‘b’. Also significant negative correlation(-.30) with all store type ‘b’.
- Majority of Stores remains closed on state holidays.
- Some Stores Shows the weekly pattern over the whole year, others show the monthly as well as fortnightly.

# Model Formulation

# Model Formulation (contd...)

- Daily sales data corresponding to three years
- Sales data may be time series data
- Also relation of sales with day, month, year (Reference EDA) indicated to apply time series model.

Questions to Answer

- Does sales data fulfills the criteria of stationarity?? If yes then
- Which model to apply
- Whether to apply same model to each store or separate model to separate store
- Identifying separate model for each store is a big challenge

# Time Series Model:

## Stationarity Check

Statistical Hypothesis Test required for checking stationarity of a time series data i.e Sales..

## Unit Root Test:

1. Augmented Dickey-Fuller(ADF) Test:

   H0(Null Hypothesis) : Data are non stationary
   H1(Alternative Hypothesis) : Data are stationary
   -Large p-values are indicative of non stationarity, and small (<0.05) p-value suggest stationarity.

# contd...

2. Kwiatkowski-Phillips-Schmidt-Shin(KPSS) Test:

   H0: Data are stationary
   H1: Data are non stationary

   -Large p-values are indicative of stationarity, and small (<0.05) p-values suggest non stationarity.

- In R, inbuilt command for both of the test is there.
- For ADF test, command is:-

  adf.test(x)

- For KPSS test, command is:-

  kpss.test(x)

  where x is time series data

# contd...

- If data are non-stationary, then can be made stationary by differencing.
- But the question is:- What is the degree of differencing??
- In R, there is command called ndiff(x) which gives the degree of differencing for time series data x.

<u>Algorithm for above Tests:</u>

*1. Do for all stores in test data*

*2. sales<-assign all sales>0 from training data set for the store*

*3. adf.test(sales)*

*4. kpss.test(sales)*

*5. ndiff(sales)*

*6. end.*

- Only the data for which Sales>0 are taken because in calculating RMSPE, day with 0 sales are ignored.

# contd...

- The results obtained suggested that there are some stores in which differencing is required.
- But we don’t know the values of p and q for fitting ARIMA(p,d,q) model
- In R, there is a command called auto.arima(x) which suggests the model in the form of ARIMA(p,d,q) for the time series data set x.

Algorithm:

*1. Do for all stores in test data*

*2. sales<-assign all sales>0 from training data set for the store*

*3. auto.arima(sales)*

*4. end.*

- Combined results are shown for the above two algorithm.

# contd...

| Stores | p_adf | p_kpss | p | d | q |
|---|---|---|---|---|---|
| 1 | 0.01 | 0.097186 | 3 | 0 | 2 |
| 3 | 0.01 | 0.1 | 5 | 0 | 5 |
| 7 | 0.01 | 0.01 | 2 | 1 | 2 |
| 8 | 0.01 | 0.01 | 5 | 1 | 5 |
| 9 | 0.01 | 0.01 | 1 | 1 | 1 |
| 10 | 0.01 | 0.01 | 2 | 1 | 2 |
| 11 | 0.01 | 0.1 | 2 | 0 | 2 |
| 12 | 0.01 | 0.022804 | 1 | 1 | 2 |
| 13 | 0.01 | 0.1 | 4 | 0 | 2 |
| 14 | 0.01 | 0.099599 | 5 | 0 | 5 |
| 15 | 0.01 | 0.013499 | 1 | 1 | 5 |
| 16 | 0.01 | 0.01 | 2 | 1 | 1 |
| 19 | 0.01 | 0.01 | 1 | 1 | 4 |
| 20 | 0.01 | 0.01 | 0 | 1 | 5 |
| 21 | 0.01 | 0.01 | 1 | 1 | 1 |
| 22 | 0.01 | 0.035108 | 2 | 1 | 5 |
| 23 | 0.01 | 0.1 | 3 | 0 | 2 |
| 24 | 0.01 | 0.01 | 3 | 1 | 5 |
| 25 | 0.01 | 0.063729 | 5 | 0 | 1 |

# contd...

- <u>Sales Prediction Algorithm:</u>

  *1. Do for all stores in test data*

  *2. sales<-assign all sales>0 from training data set for the store*

  *3. fit <- auto.arima(sales)*

  *4. forecast(fit, h=no. of days required for prediction for each store)*

  *5. end.*
- Obtained RMSPE is 0.28311 which is not so good.
- Time series model alone is not sufficient for capturing all the variability in data.

<u>Problems in the previous model:</u>

- Features are not taken into account.
- Even after differencing with the suggested d value, data might not be stationary.

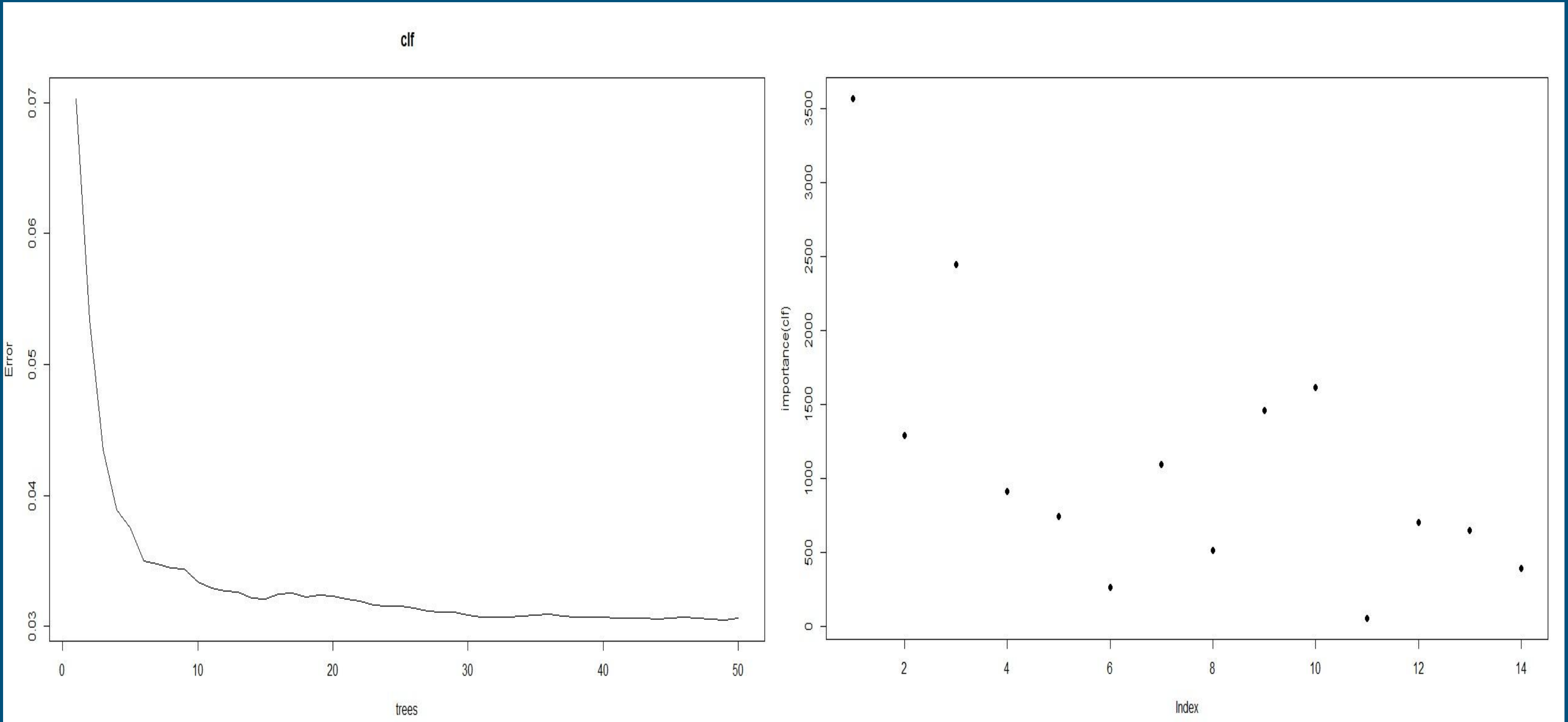
# Random Forest Model

- Almost all features are categorical
- Response variable (sales) is continuous
- Model which can classify and fit regression simultaneously is needed
- Random forest is applied on all the stores as a whole.

First Model using Random Forest

- Rows having 0 sales are deleted
- Date feature is splitted into corresponding days, months, years
- Train Table and Store tables are merged
- Features used in Random Forest(Feature Number): 1. Store, 2. Dayofweek, 3. Promo, 4. day, 5. month, 6. year, 7. StoreType, 8. Assortment, 9. CompetitionOpenSinceMonth, 10. CompetitionOpenSinceYear, 11. Promo2, 12. Promo2SinceWeek, 13. Promo2SinceYear, 14. PromoInterval

# Error v/s No of Trees and GINI Index Plot for Importance of Variables.



# Results with Random Forest

Prediction Result:

- Obtained RMSPE is 0.16582 better than time series model.

Modification in Application of Random Forest

- Now apply Random Forest on each store data  separately
- Features remains the same as before

Prediction Result:

- Obtained RMSPE is 0.11509 which is better than applying as a whole.

# Status at Kaggle Leaderboard after Random Forest

- Rank is 1357 and RMPSE is 0.11509

| | | | | | |
|---|---|---|---|---|---|
| 1350 | ↓121 | Albert Wong | 0.11503 | 5 | Wed, 18 Nov 2015 18:51:26 (-30.6d) |
| 1351 | ↓121 | Haik | 0.11505 | 21 | Mon, 26 Oct 2015 11:04:12 (-10.8d) |
| 1352 | ↓121 | somendra tripathi | 0.11506 | 3 | Mon, 02 Nov 2015 08:47:23 |
| 1353 | ↓121 | vivi.chen | 0.11507 | 5 | Sun, 11 Oct 2015 01:56:18 |
| 1354 | ↑643 | OP | 0.11507 | 7 | Sat, 28 Nov 2015 18:17:37 (-8.6h) |
| 1355 | ↓122 | Gennady | 0.11508 | 4 | Sat, 14 Nov 2015 23:56:44 (-36.1d) |
| 1356 | ↓122 | LokeshPanchariya | 0.11509 | 3 | Sun, 18 Oct 2015 13:11:48 (-2.7h) |
| **1357** | **↓122** | **sethsachin21** | **0.11509** | **14** | **Wed, 02 Dec 2015 12:28:35 (-19.6d)** |
| 1358 | ↓122 | Yuval Harpaz | 0.11509 | 5 | Thu, 22 Oct 2015 11:06:01 |
| 1359 | ↓122 | lorifei | 0.11513 | 1 | Mon, 19 Oct 2015 00:18:57 |
| 1360 | ↓122 | Antoine Miech | 0.11514 | 30 | Sat, 07 Nov 2015 12:14:04 (-27.8d) |
| 1361 | ↓122 | omgponies | 0.11516 | 14 | Thu, 29 Oct 2015 10:50:21 (-20.8d) |
| 1362 | new | wenyu | 0.11517 | 5 | Wed, 02 Dec 2015 11:46:30 (-22.9h) |

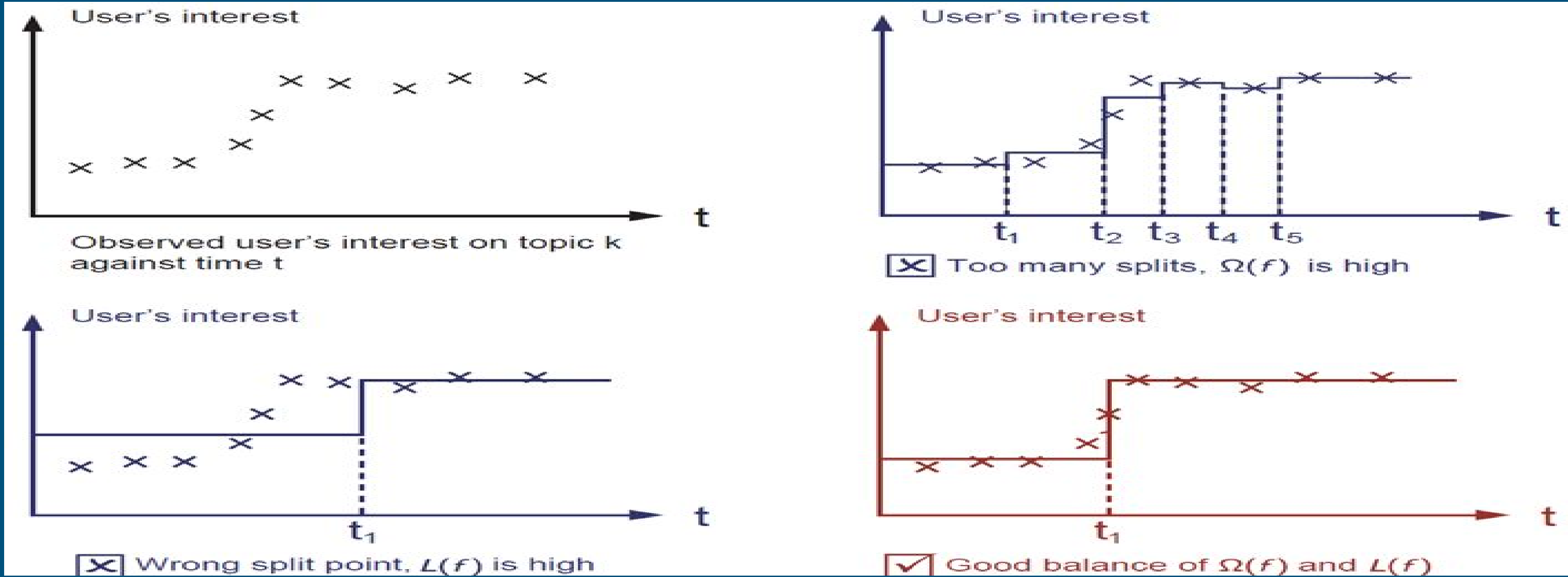
# XGBoost Model

- Random forest uses bootstrapping method for training while gradient boosting builds decision tree over the residuals .
- The final prediction is also not a simple average but the weighted average.
- Random forest uses decision tree for prediction while in gradient boosting it could be decision tree or KNN or SVM.

# Objective function of XGBoost

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta)$$

- 1st term of objective function is training loss function which measures how predictive our model is.
- 2nd term is regularization term which helps us to overfitting the data.



# Prediction Result

- Obtained RMPSE is 0.127830
- Better results when take weighted average of Random Forest result and XGBoost result and obtained RMPSE is 0.11154
- This is the best result which we got till. Our team rank is 1213 out of total 3242.
- Our RMSPE is 0.02095 more than the first in leaderboard score.

| | | | | | |
|---|---|---|---|---|---|
| 1173 | ↓119 | Prabhudatta Das | 0.11144 | 18 | Fri, 06 Nov 2015 07:27:28 (-12.6d) |
| 1174 | ↓119 | magenta | 0.11145 | 17 | Wed, 04 Nov 2015 22:56:58 (-24.8d) |
| 1175 | ↓119 | Sayantan Raha | 0.11146 | 36 | Wed, 25 Nov 2015 16:58:27 (-27.1d) |
| 1176 | ↓119 | Indigo | 0.11149 | 15 | Thu, 05 Nov 2015 19:08:47 (-7.8d) |
| **1177** | **↑59** | **sethsachin21** | **0.11154** | **18** | **Wed, 02 Dec 2015 19:42:36** |

**Your Best Entry ↑**
You improved on your best score by 0.00031.
You just moved up 14 positions on the leaderboard. Tweet this!

| | | | | | |
|---|---|---|---|---|---|
| 1178 | ↓119 | JohnMitchell | 0.11157 | 2 | Fri, 09 Oct 2015 18:13:49 |
| 1179 | ↓119 | Andrea Palaia | 0.11158 | 33 | Thu, 12 Nov 2015 10:53:45 (-10.2d) |
| 1180 | ↓119 | DenverDeo | 0.11160 | 13 | Fri, 06 Nov 2015 15:49:22 (-13.3d) |
| 1181 | ↓119 | Sean Maybee | 0.11163 | 52 | Sun, 15 Nov 2015 23:00:06 (-30.9d) |
| 1182 | ↓118 | Prateek Singhal | 0.11167 | 17 | Wed, 14 Oct 2015 07:22:10 (-8.7d) |

# Challenges And Learnings



# Continued...

## Challenges:

1) Handling large amount of  sales data (10,17,210 observations on 13 variables)
2) Some 180 stores were closed for 6 months. Unable to fill the gap of sales for those stores..
3) Prediction of sales for individual stores(out of 1115) and most of stores have different pattern of sales. A single model cannot fit to all stores.

## Learnings:

1) Exploring large datasets using visualisation tools.
2) Learn the application of Time Series, Random Forest, XG Boost.

# Scope of Improvement

- Applied only three algorithms i.e Time Series algo, random forest and XGBoost. So there are scope for applying more algorithms like time series linear models, KNN Regression, Unobserved Component Model, Principal Component Regression.
- By taking the regression of all the models for all the sales data may predict the sales better. In our case weighted average of Random Forest output and XGBoost gives better result than individual algorithms.

# References


- Kaggle competition Forum
- https://www.otexts.org/fpp
- Discussion with batchmates(Robin Singh)
- https://xgboost.readthedocs.org/en/latest/
- https://www.kaggle.com/wiki/RandomForests

# Thank You