04-10-2020

RETAIL DATA ANALYSIS PROJECT

CAPSTONE PROJECT

PRESENTED BY: SHILPA GIRIDHAR

CONTENTS

PROJECT NOTES 1 – EDA

DESCRIPTION

The data set provides historical sales data from 2010-02-05 to 2012-11-01 for 45 stores located in different regions - each store contains a number of departments.

The company also runs several promotional markdown events throughout the year. These markdowns precede prominent holidays, the four largest of which are **the Super Bowl**, **Labor Day**, **Thanksgiving**, and **Christmas**. The weeks including these holidays are weighted five times higher in the evaluation than non-holiday weeks.

Within this dataset you will find the following fields:

1	Store	the store number
2	Dept	the department number
3	Date	the week
4	IsHoliday	whether the week is a special holiday week
5	Type	the type of store
6	Size	the size of store
7	Weekly_Sales	sales for the given department in the given store
8	Temperature	average temperature in the region
9	Fuel_Price	cost of fuel in the region
10	MarkDown1-5	anonymized data related to promotional markdowns. MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA
11	CPI	the consumer price index
12	Unemployment	the unemployment rate

OBJECTIVE

- Predict the department-wide sales for each store for the following year
- Model the effects of markdowns on holiday weeks
- Provide recommended actions based on the insights drawn, with prioritization placed on largest business impact.

The Challenge - One challenge of modeling retail data is the need to make decisions based on limited history. Holidays and select major events come once a year, and so does the chance to see how strategic decisions impacted the bottom line. In addition, markdowns are known to affect sales – the challenge is to predict which departments will be affected and to what extent.

REQUIREMENTS

Project Notes- 1 Expectations:
1) Business Problem Understanding and Problem definition
2) Generate a data report.
3) Exploratory Data analysis and insights driven from it.

DATA EXPLORATION

DATA TYPE OF ALL VARAIBLES

The data shows that there are

- 8190 observations and 12 variables in Features Dataset
- 45 observations and 3 variables in Stores Dataset
- 421570 observations and 5 variables in Sales Dataset

We perform a left join the 3 datasets using the “Store” Column to get a combined dataset with

- 421570 observations and 16 variables.

```
tibble [421,570 x 16] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Store       : num [1:421570] 1 1 1 1 1 1 1 1 1 1 ...
 $ Date        : chr [1:421570] "01/04/2011" "01/04/2011" "01/04/2011" "01/04/2011" ...
 $ IsHoliday   : logi [1:421570] FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ Dept        : num [1:421570] 49 26 81 34 59 30 7 85 8 28 ...
 $ Weekly_Sales: num [1:421570] 13168 5947 28545 9950 317 ...
 $ Type        : chr [1:421570] "A" "A" "A" "A" ...
 $ Size        : num [1:421570] 151315 151315 151315 151315 151315 ...
 $ Temperature : num [1:421570] 59.2 59.2 59.2 59.2 59.2 ...
 $ Fuel_Price  : num [1:421570] 3.52 3.52 3.52 3.52 3.52 ...
 $ MarkDown1   : num [1:421570] NA NA NA NA NA NA NA NA NA NA ...
 $ MarkDown2   : num [1:421570] NA NA NA NA NA NA NA NA NA NA ...
 $ MarkDown3   : num [1:421570] NA NA NA NA NA NA NA NA NA NA ...
 $ MarkDown4   : num [1:421570] NA NA NA NA NA NA NA NA NA NA ...
 $ MarkDown5   : num [1:421570] NA NA NA NA NA NA NA NA NA NA ...
 $ CPI         : num [1:421570] 215 215 215 215 215 ...
 $ Unemployment: num [1:421570] 7.68 7.68 7.68 7.68 7.68 ...
 - attr(*, "spec")=
  .. cols(
  ..   Store = col_double(),
  ..   Date = col_character(),
  ..   IsHoliday = col_logical(),
  ..   Dept = col_double(),
  ..   Weekly_Sales = col_double(),
  ..   Type = col_character(),
  ..   Size = col_double(),
  ..   Temperature = col_double(),
  ..   Fuel_Price = col_double(),
  ..   MarkDown1 = col_double(),
  ..   MarkDown2 = col_double(),
  ..   MarkDown3 = col_double(),
  ..   MarkDown4 = col_double(),
  ..   MarkDown5 = col_double(),
  ..   CPI = col_double(),
  ..   Unemployment = col_double()
  .. )
```

A summary on the dataset shows that columns "Markdown 1-5" have NAs.

Date column needs to be parsed as Date class type.

The Store, Dept, and Type columns need to be parsed as factor type. After parsing the columns, we get a summary of the dataset as follows -

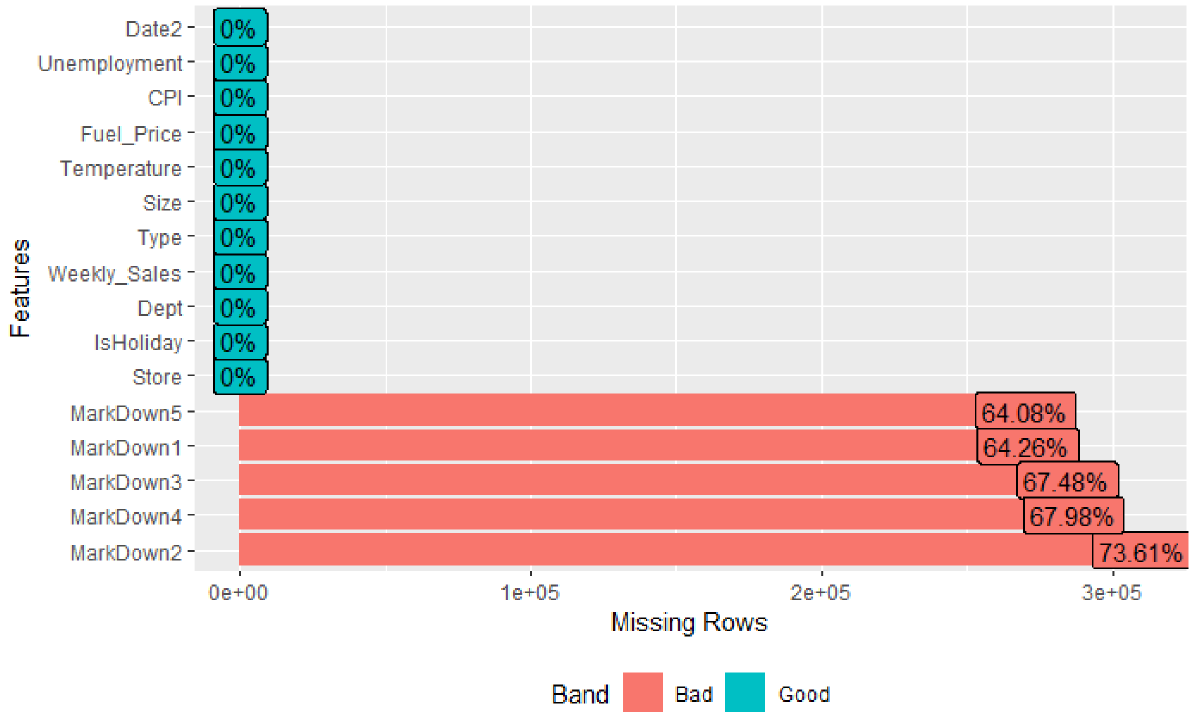
```
     Store          IsHoliday            Dept        Weekly_Sales     Type            Size       
 13      : 10474   Mode :logical   Min.   : 1.00   Min.   : -4989   A:215478   Min.   : 34875  
 10      : 10315   FALSE:391909    1st Qu.:18.00   1st Qu.:  2080   B:163495   1st Qu.: 93638  
 4       : 10272   TRUE :29661     Median :37.00   Median :  7612   C: 42597   Median :140167  
 1       : 10244                   Mean   :44.26   Mean   : 15981              Mean   :136728  
 2       : 10238                   3rd Qu.:74.00   3rd Qu.: 20206              3rd Qu.:202505  
 24      : 10228                   Max.   :99.00   Max.   :693099              Max.   :219622  
 (Other):359799                                                                                
  Temperature       Fuel_Price      MarkDown1          MarkDown2           MarkDown3       
 Min.   : -2.06   Min.   :2.472   Min.   :    0.27   Min.   :  -265.8   Min.   :   -29.10  
 1st Qu.: 46.68   1st Qu.:2.933   1st Qu.: 2240.27   1st Qu.:    41.6   1st Qu.:     5.08  
 Median : 62.09   Median :3.452   Median : 5347.45   Median :   192.0   Median :    24.60  
 Mean   : 60.09   Mean   :3.361   Mean   : 7246.42   Mean   :  3334.6   Mean   :  1439.42  
 3rd Qu.: 74.28   3rd Qu.:3.738   3rd Qu.: 9210.90   3rd Qu.:  1926.9   3rd Qu.:   103.99  
 Max.   :100.14   Max.   :4.468   Max.   :88646.76   Max.   :104519.5   Max.   :141630.61  
                                  NA's   :270889     NA's   :310322     NA's   :284479     
   MarkDown4          MarkDown5             CPI         Unemployment        Date2           
 Min.   :    0.22   Min.   :   135.2   Min.   :126.1   Min.   : 3.879   Min.   :2010-02-05  
 1st Qu.:  504.22   1st Qu.:  1878.4   1st Qu.:132.0   1st Qu.: 6.891   1st Qu.:2010-10-08  
 Median : 1481.31   Median :  3359.4   Median :182.3   Median : 7.866   Median :2011-06-17  
 Mean   : 3383.17   Mean   :  4629.0   Mean   :171.2   Mean   : 7.960   Mean   :2011-06-18  
 3rd Qu.: 3595.04   3rd Qu.:  5563.8   3rd Qu.:212.4   3rd Qu.: 8.572   3rd Qu.:2012-02-24  
 Max.   :67474.85   Max.   :108519.3   Max.   :227.2   Max.   :14.313   Max.   :2012-10-26  
 NA's   :286603     NA's   :270138                                                         
```

- Data Type of each variable

$Store [1] "factor"	$IsHoliday [1] "logical"
$Dept [1] "numeric"	$Weekly_Sales [1] "numeric"
$Type [1] "factor"	$Size [1] "numeric"
$Temperature [1] "numeric"	$Fuel_Price [1] "numeric"
$MarkDown1 [1] "numeric"	$MarkDown2 [1] "numeric"
$MarkDown3 [1] "numeric"	$MarkDown4 [1] "numeric"
$MarkDown5 [1] "numeric"	$CPI [1] "numeric"
$Unemployment [1] "numeric"	$Date2 [1] "Date"

Missing Values and Outlier Treatment

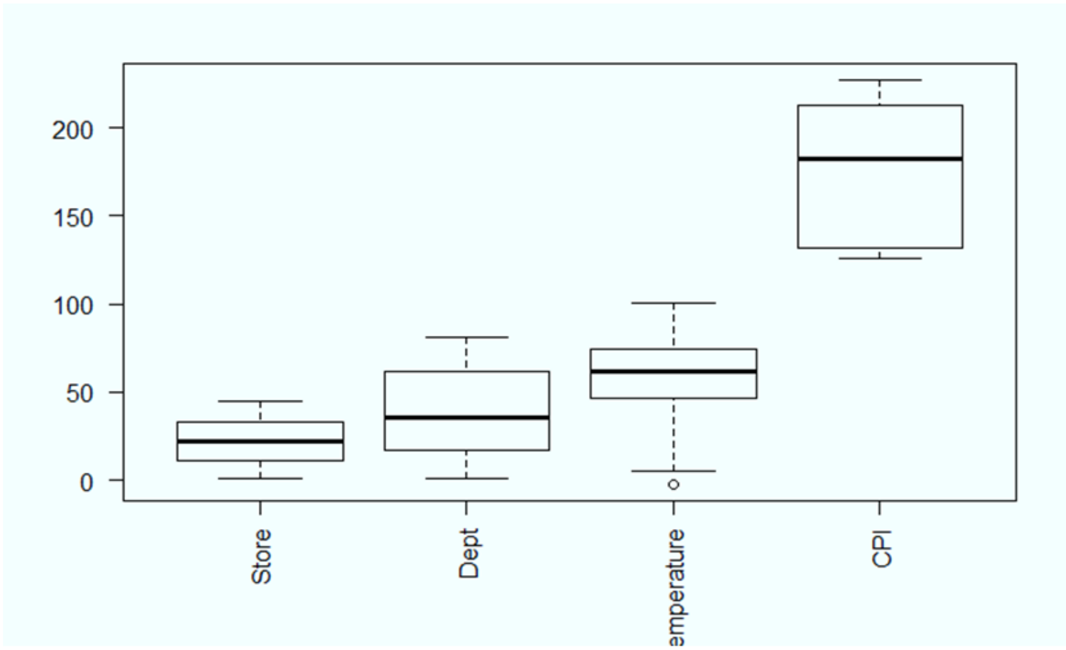
- Missing Data Plot - Used the “library(DataExplorer)” and use the function “plot_missing()” to ascertain that the missing data; Markdown 1 to 5 column show missing data.
- MarkDowns columns with NA's or negative numbers will be replaced by zeros to overcome the missing value data problems.

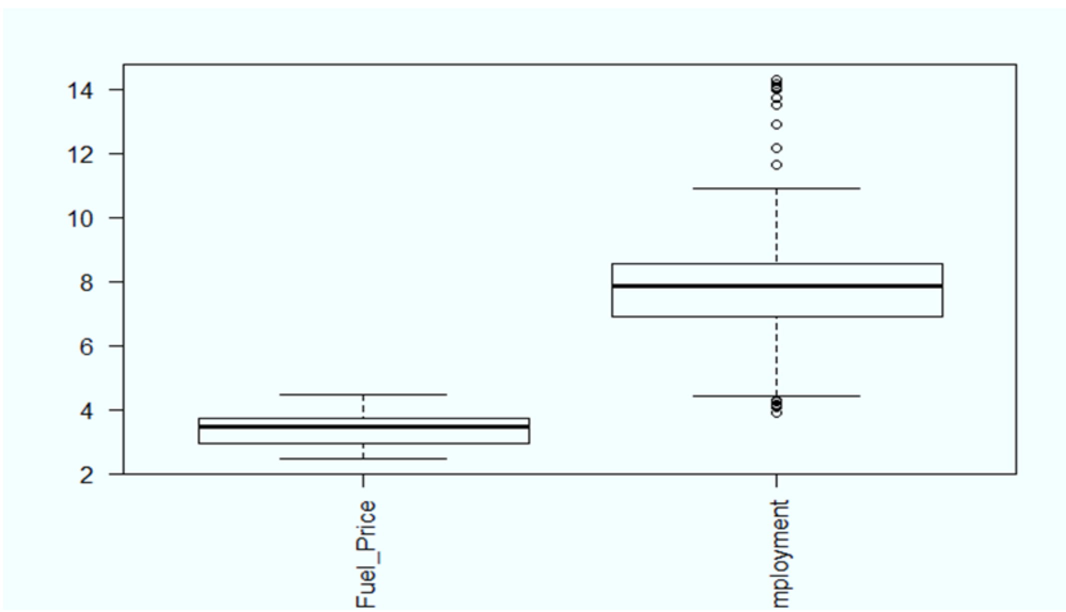


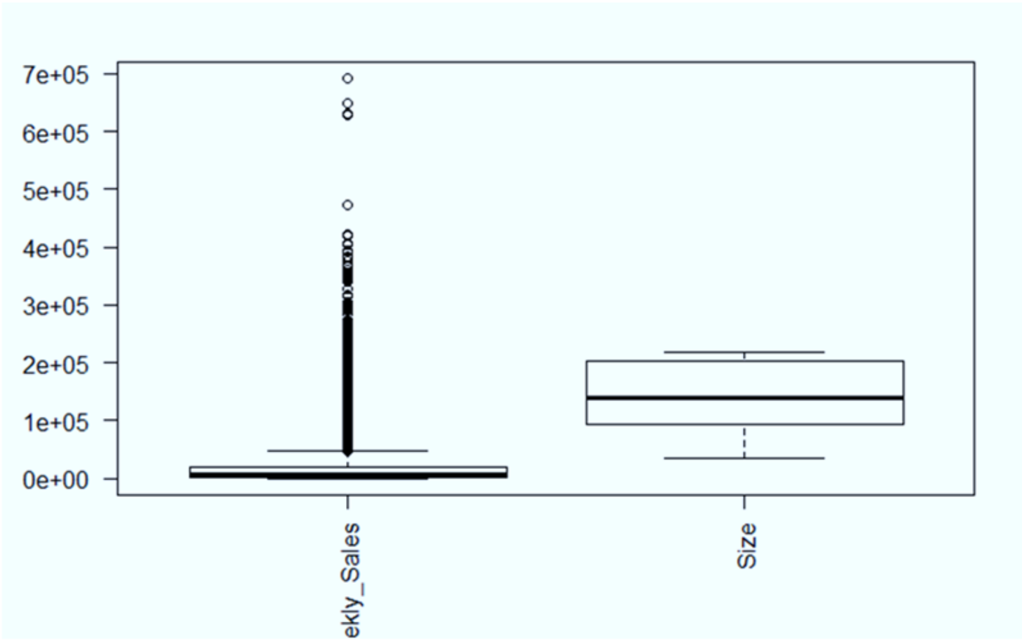
- Detect and treat outliers for non holiday records

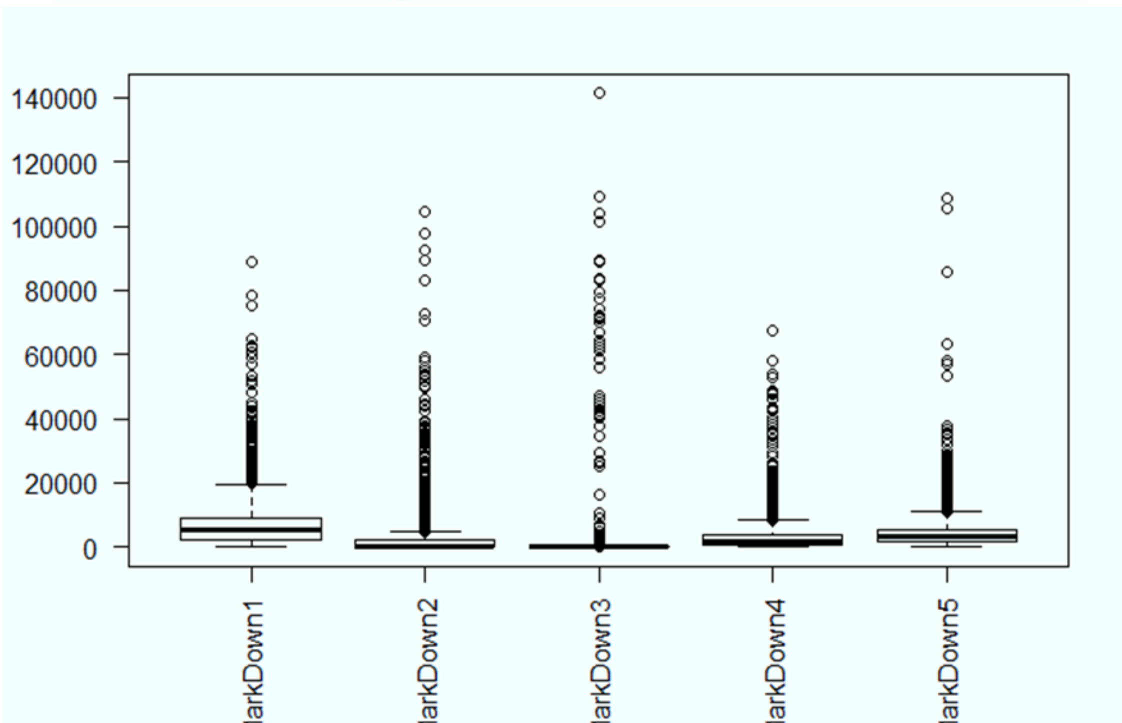
```
outlier_capping = function(x){
 qnt = quantile(x, probs=c(.25, .75), na.rm = T)
 caps = quantile(x, probs=c(.05, .95), na.rm = T)
  H = 1.5 * IQR(x, na.rm = T)
  x[x < (qnt[1] - H)] <- caps[1]
  x[x > (qnt[2] + H)] <- caps[2]
  return(x) }
df$Weekly_Sales=outlier_capping(df$Weekly_Sales)
df$MarkDown1=outlier_capping(df$MarkDown1)
df$MarkDown2=outlier_capping(df$MarkDown2)
df$MarkDown3=outlier_capping(df$MarkDown3)
df$MarkDown4=outlier_capping(df$MarkDown4)
df$MarkDown5=outlier_capping(df$MarkDown5)
```

- Individual box plots after outlier treatment in Weekly Sales and Markdown columns






7e+05
6e+05
5e+05
4e+05
3e+05
2e+05
1e+05
0e+00
ekly_Sales
Size

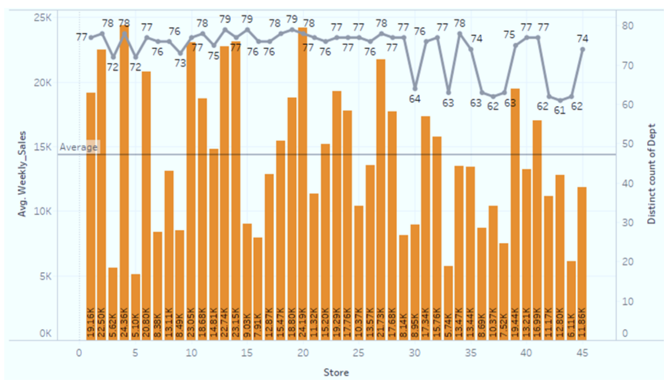

140000
120000
100000
80000
60000
40000
20000
0
/larkDown1
/larkDown2
/larkDown3
/larkDown4
/larkDown5

UNIQUE VALUES IN EACH FEATURE

- How many stores, department, store types are present in data?
- Stores – 45
- Department – 81
- Store Types – 3 (noted as A,B, C) based on the size of the store
- Dates ranging from 2010-02-05 to 2012-11-01
- Markdown 1 to 5 indicate certain discounts and this data is available only from Nov-2011
- Aggregate Functions to summarize different variables
- Weekly Sales vs Type of Store

Type <fctr>	x <dbl>
A	4331014723
B	2000700737
C	405503528

3 rows

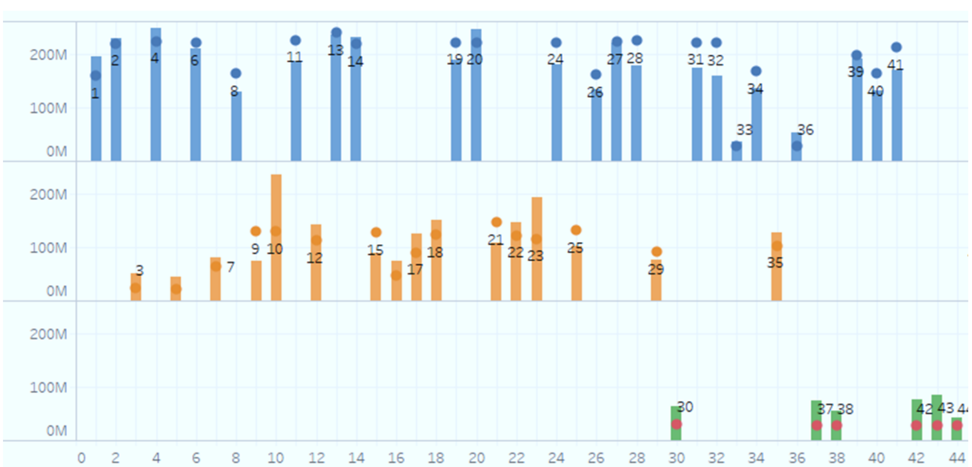


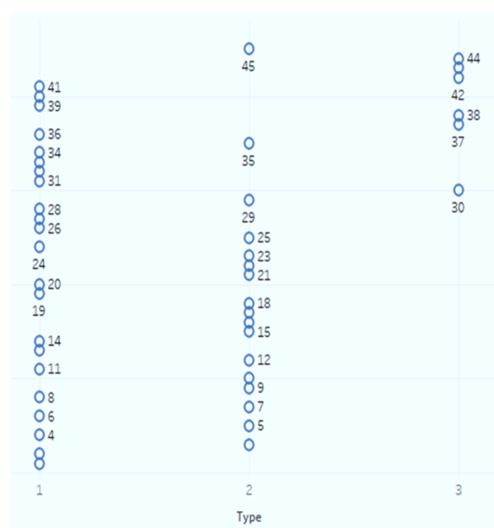
22 Stores in Type 1, 17 Stores of Type 2, 6 Stores of Type 3

Each Store has number of departments ranging from 60 -70

Average Sales across stores is about ~ $14K

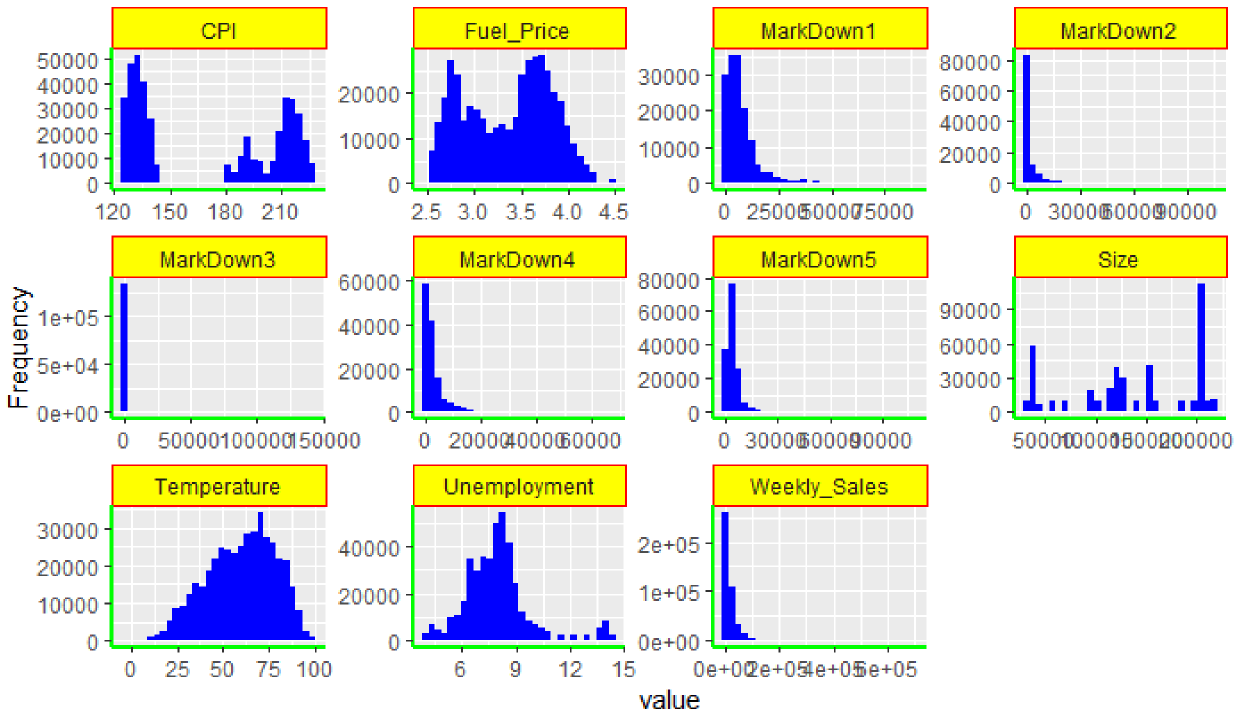
Quite evident that Sales is proportional to size of the store





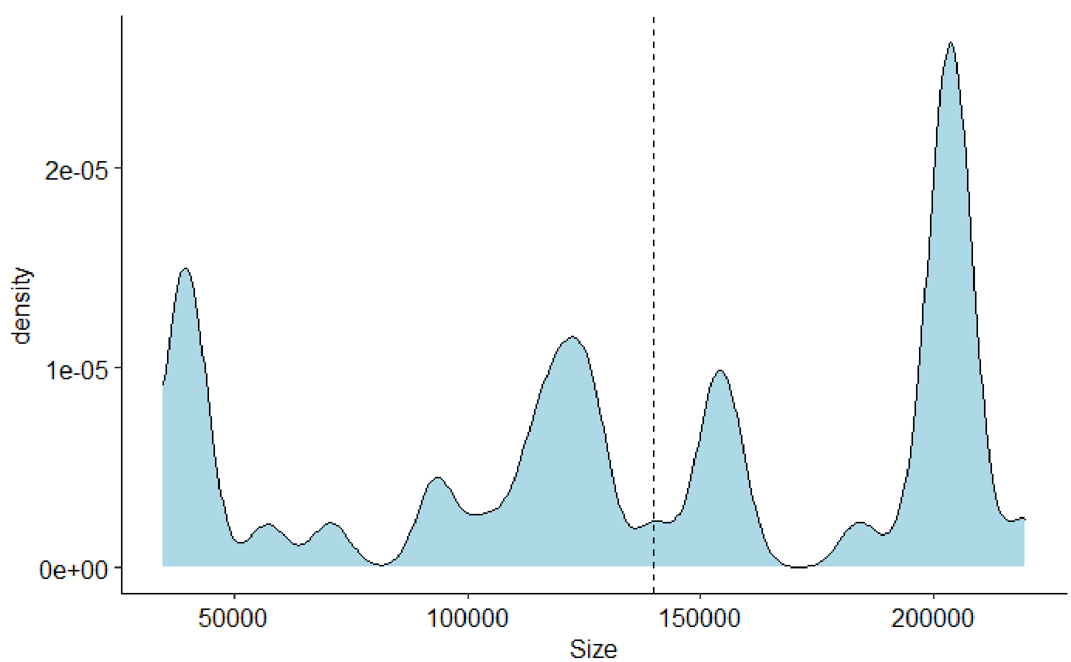
UNIVARIATE ANALYSIS

- Histogram plot to understand continuous variables

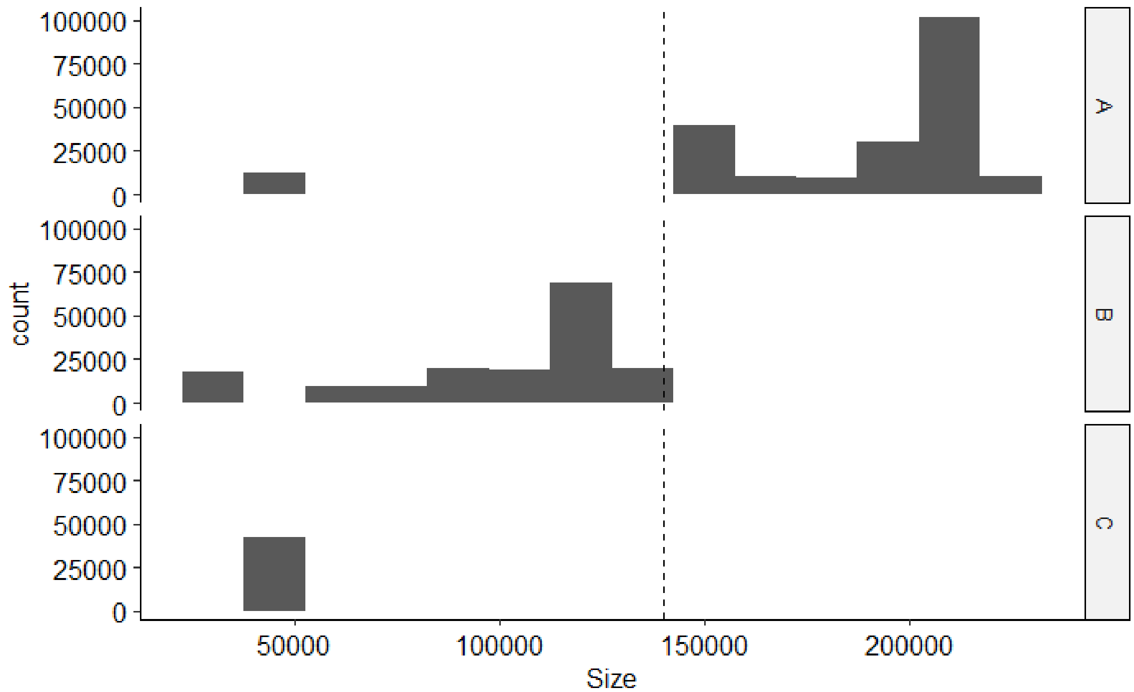


- Size of the Stores

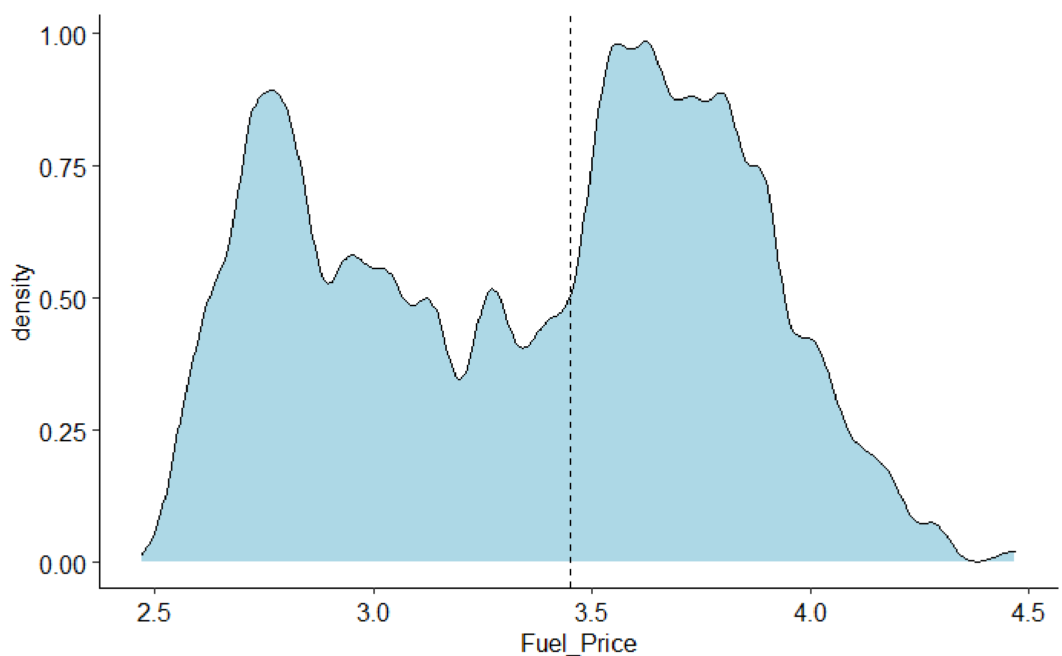
Density Plot



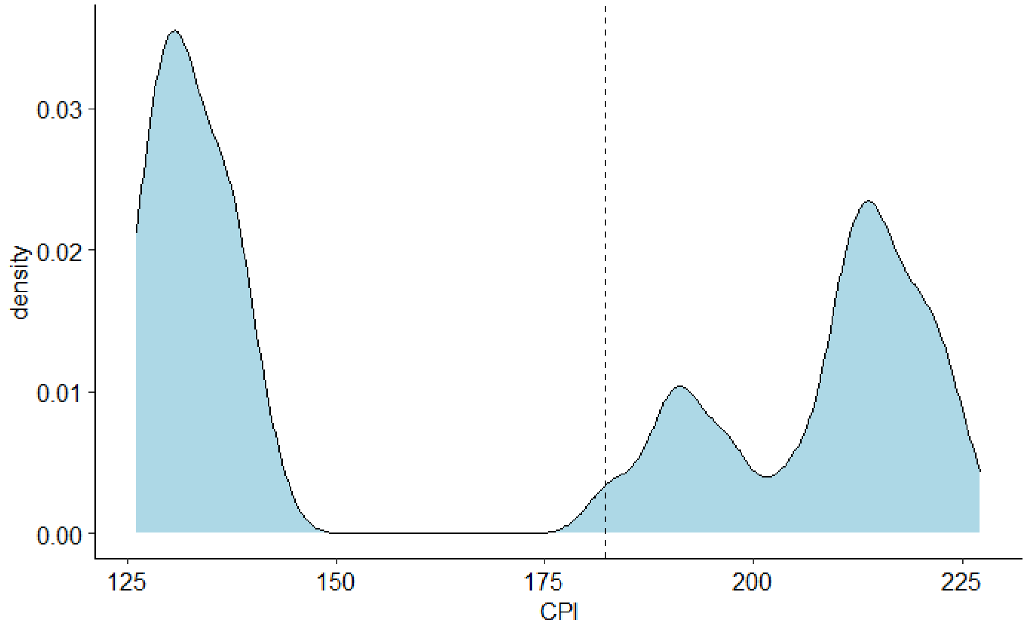
Store Size per Store Type of Store Histogram plot



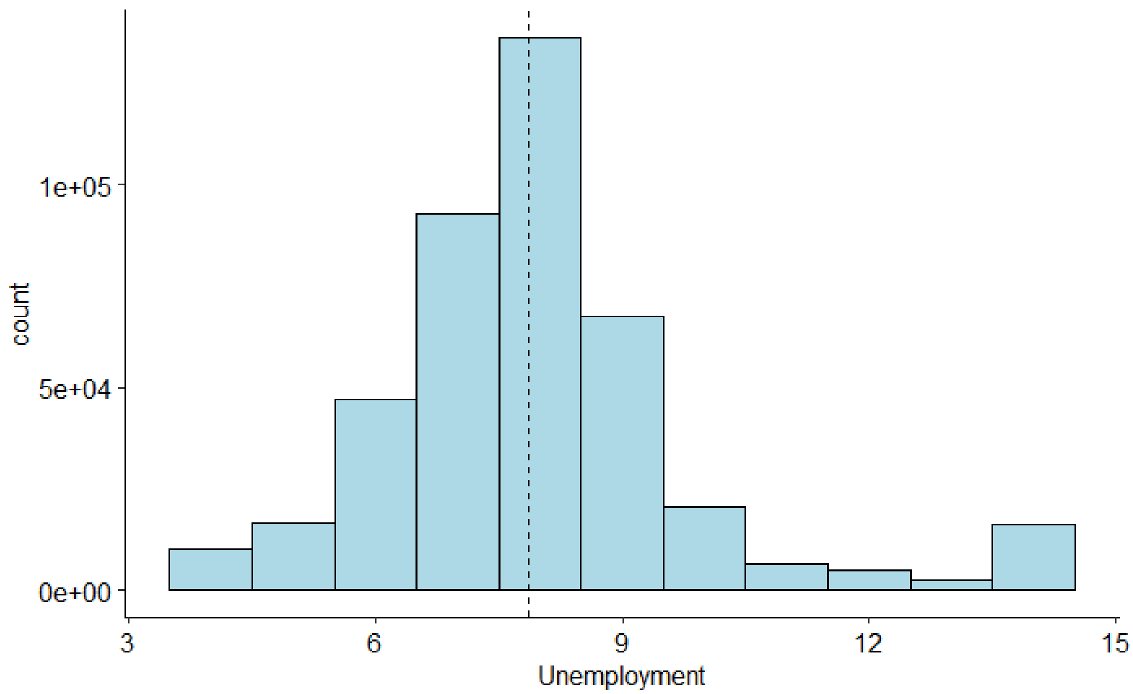
- Fuel Price distribution



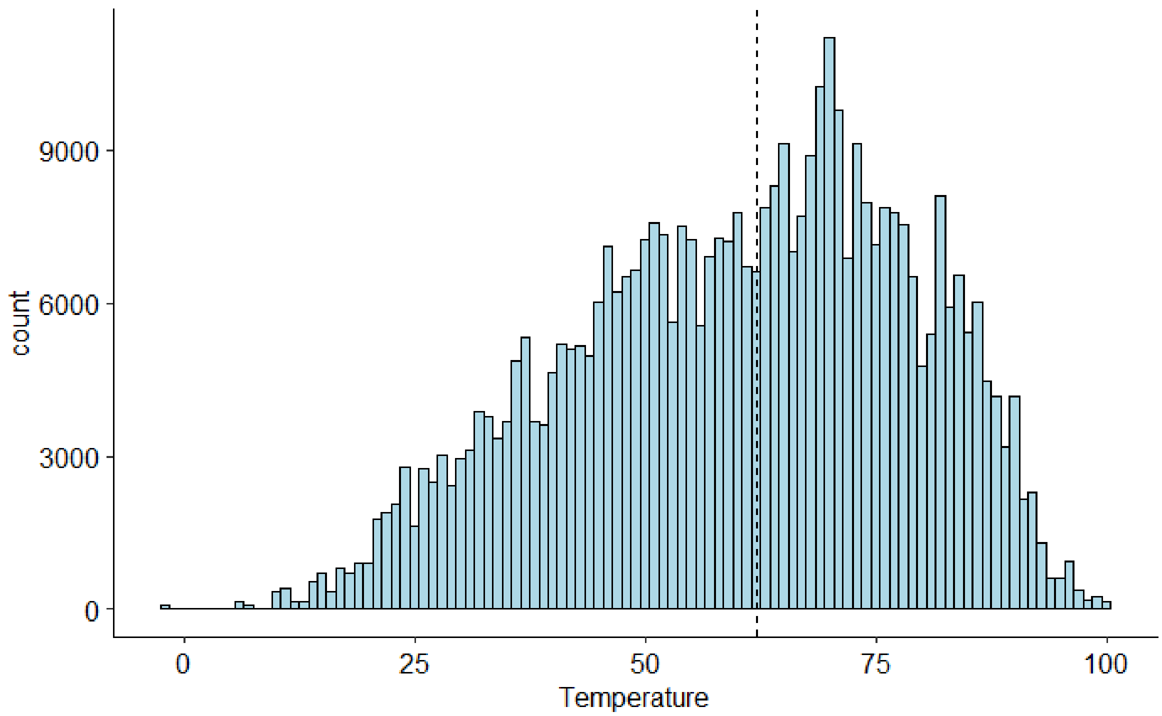
- CPI distribution



- Unemployment distribution

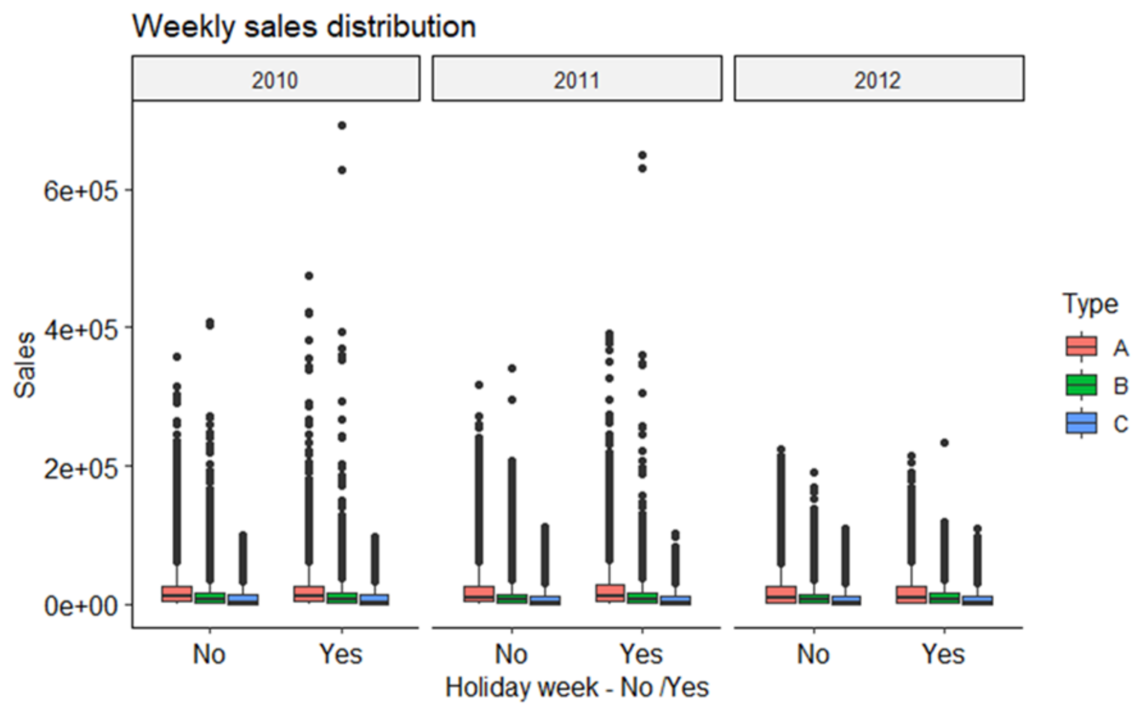


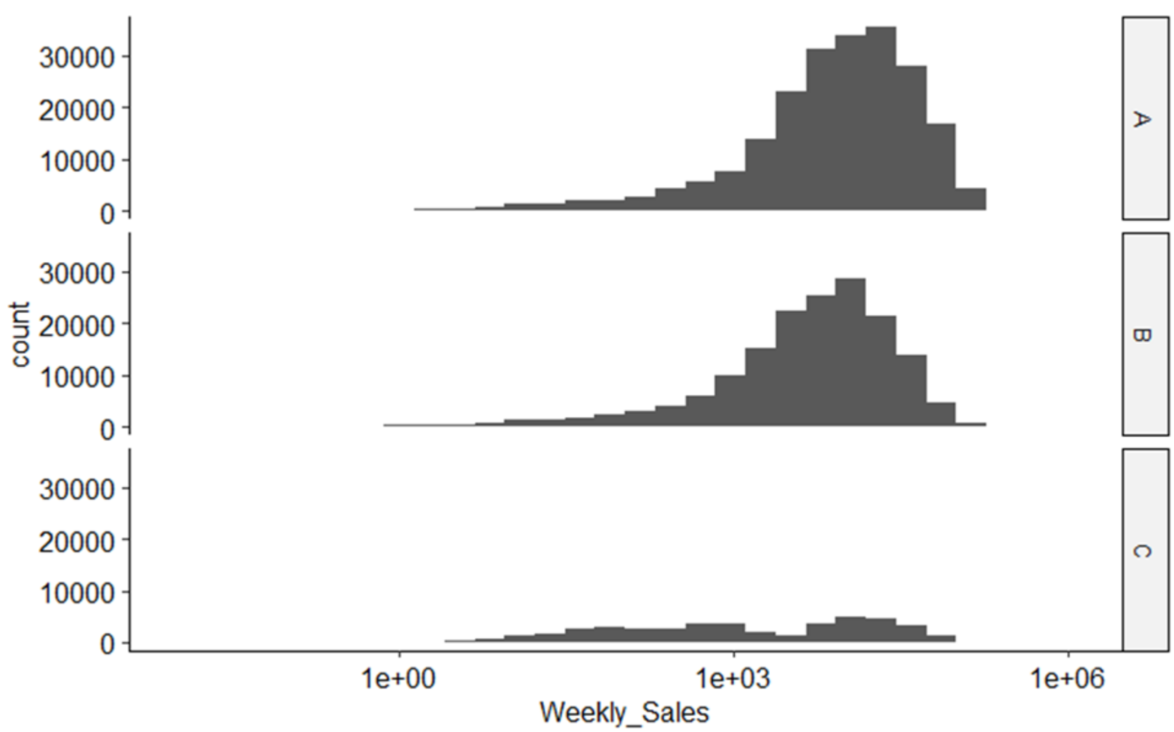
- Temperature Distribution



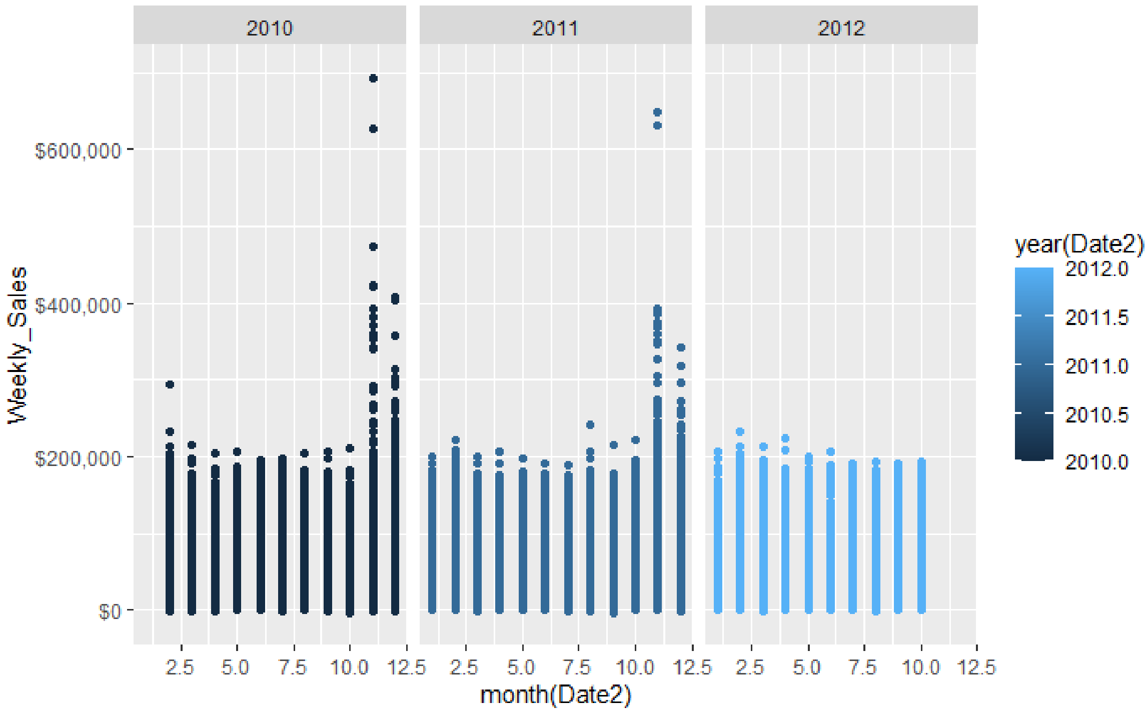
BIVARIATE ANALYSIS

- “Weekly Sales” vs “Store Type”; From this plot, we notice that type C stores have fewer sales. But both Type A and B stores have outliers.

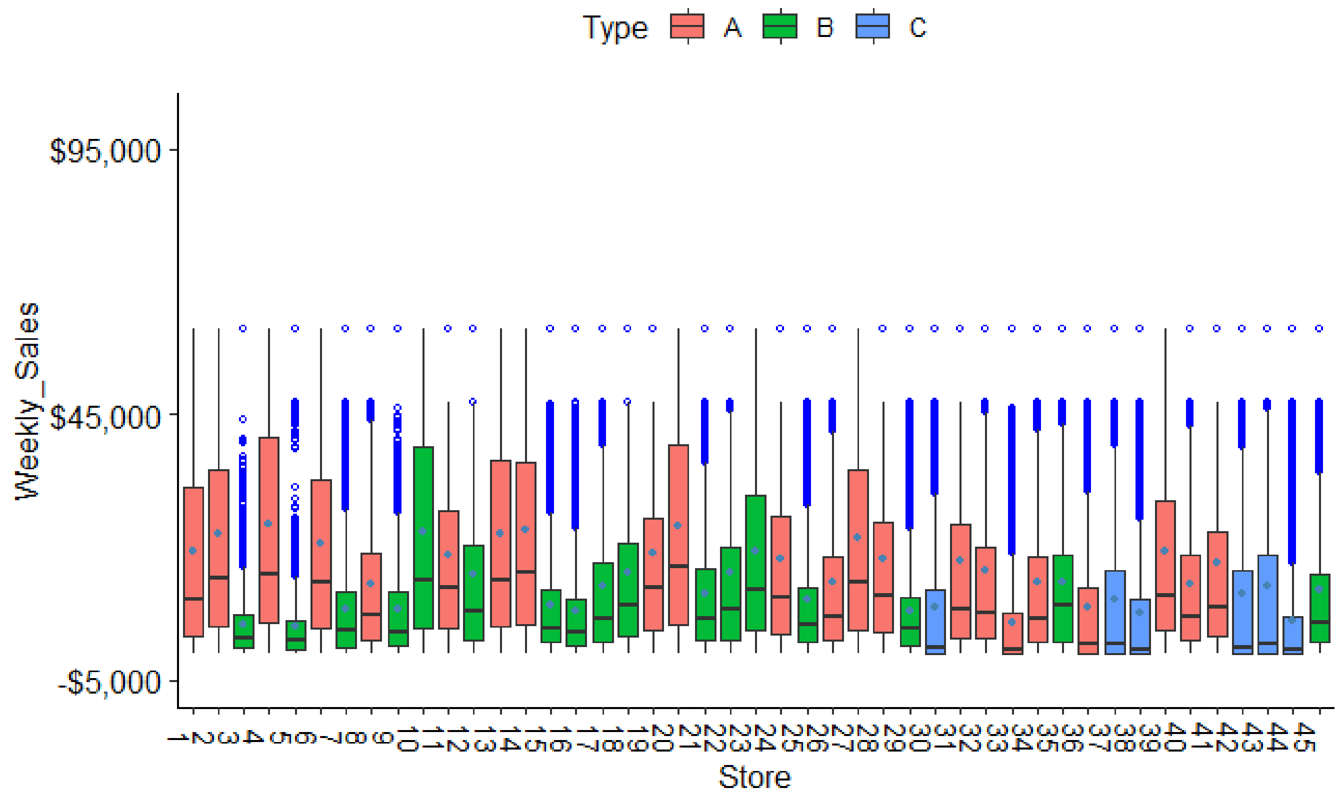


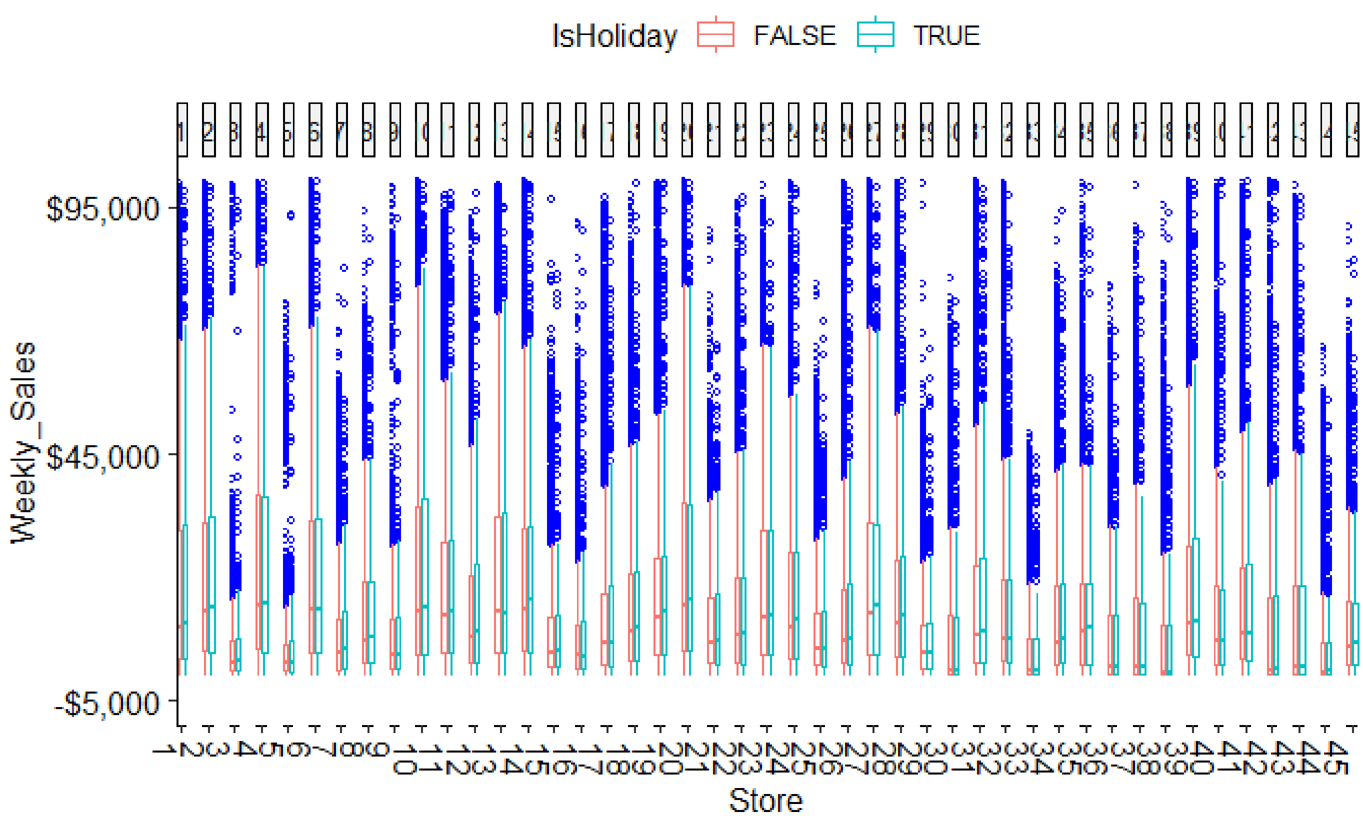


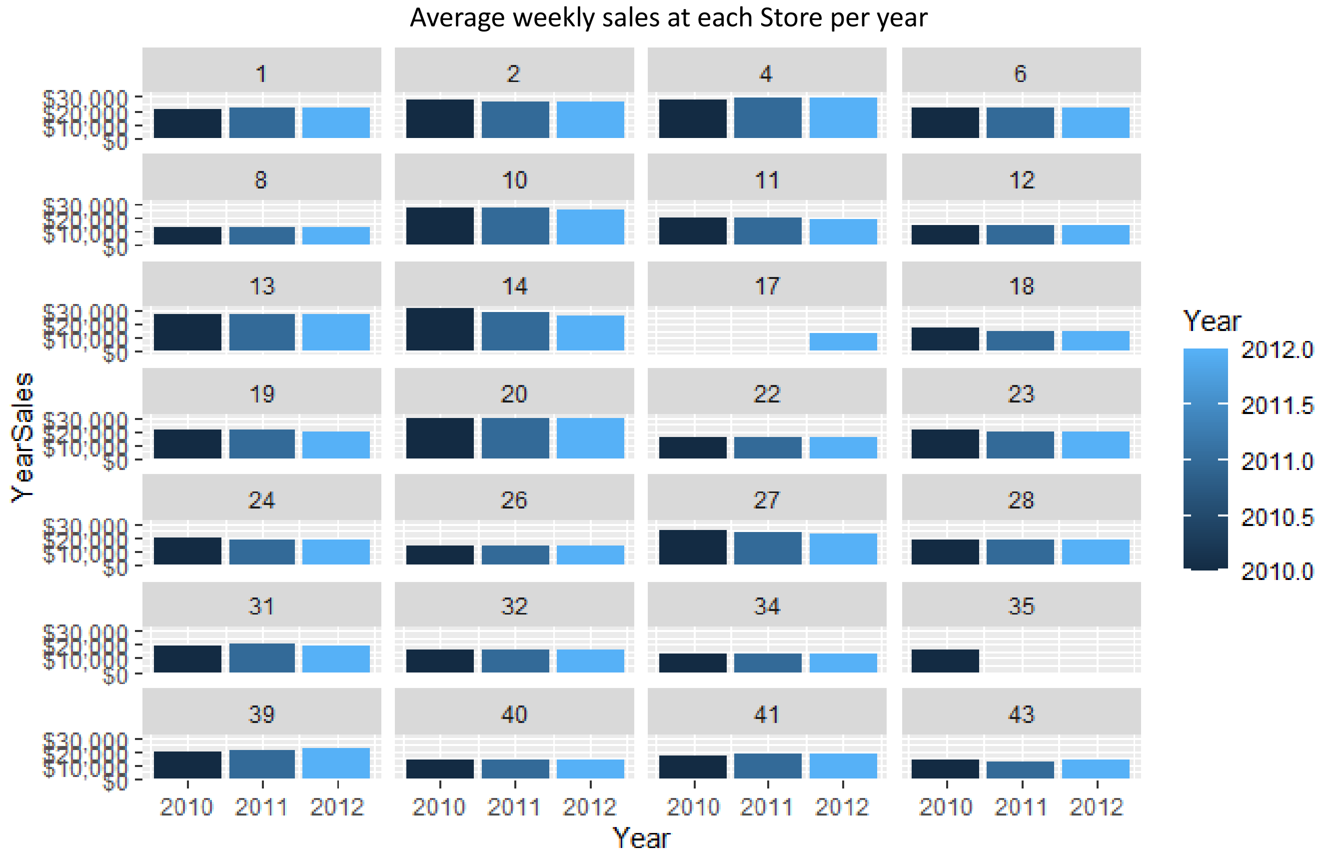
- "Weekly Sales" vs "Month of the Year" and "Holiday Week" It shows that weekly sales volume peak during certain weeks of the year.



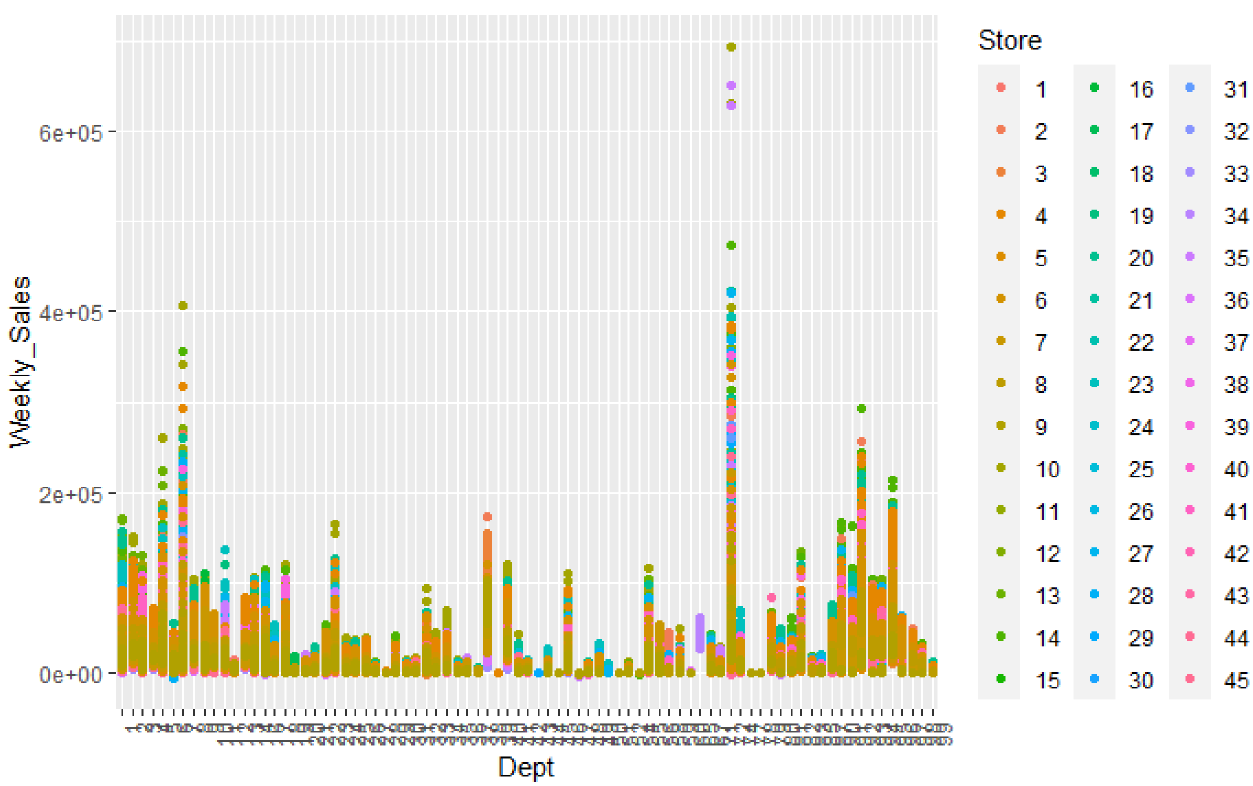
- Plot of Sales Distribution by Stores

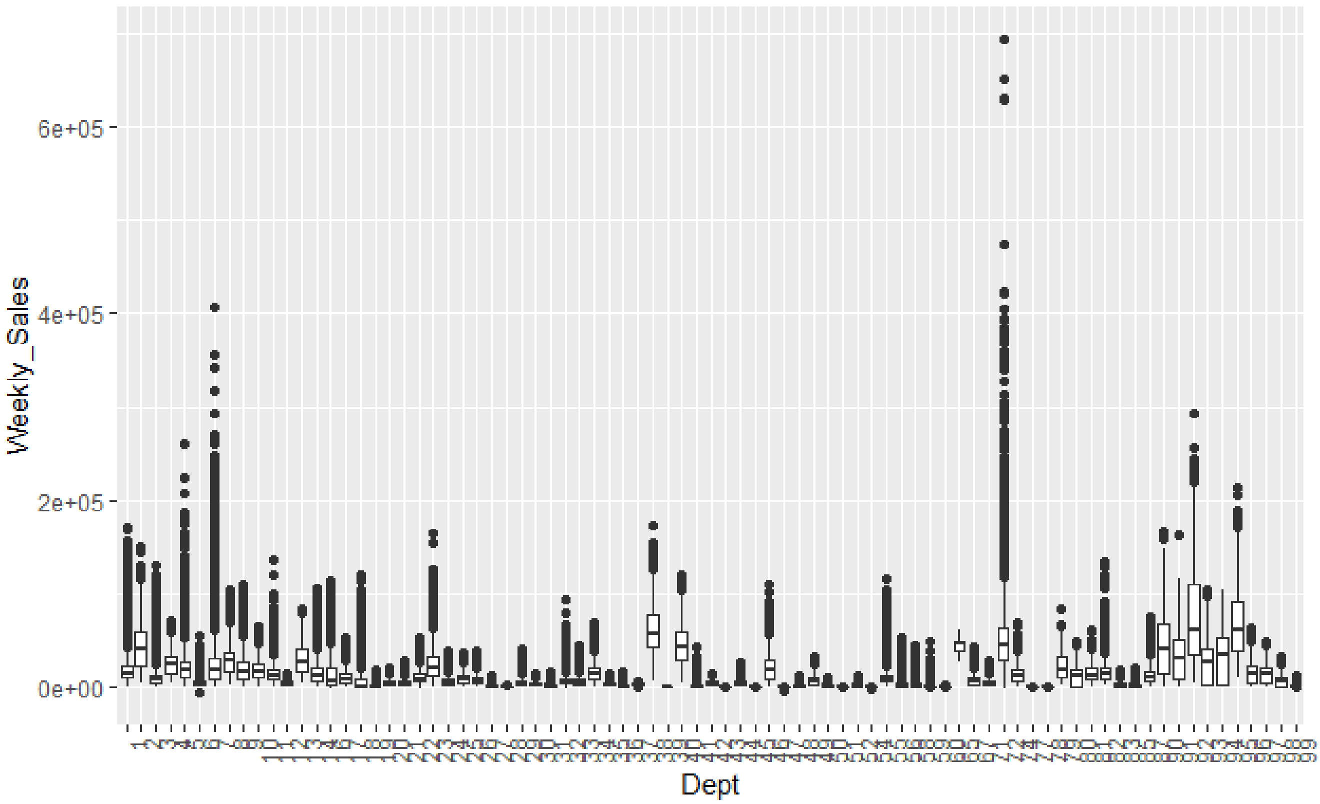




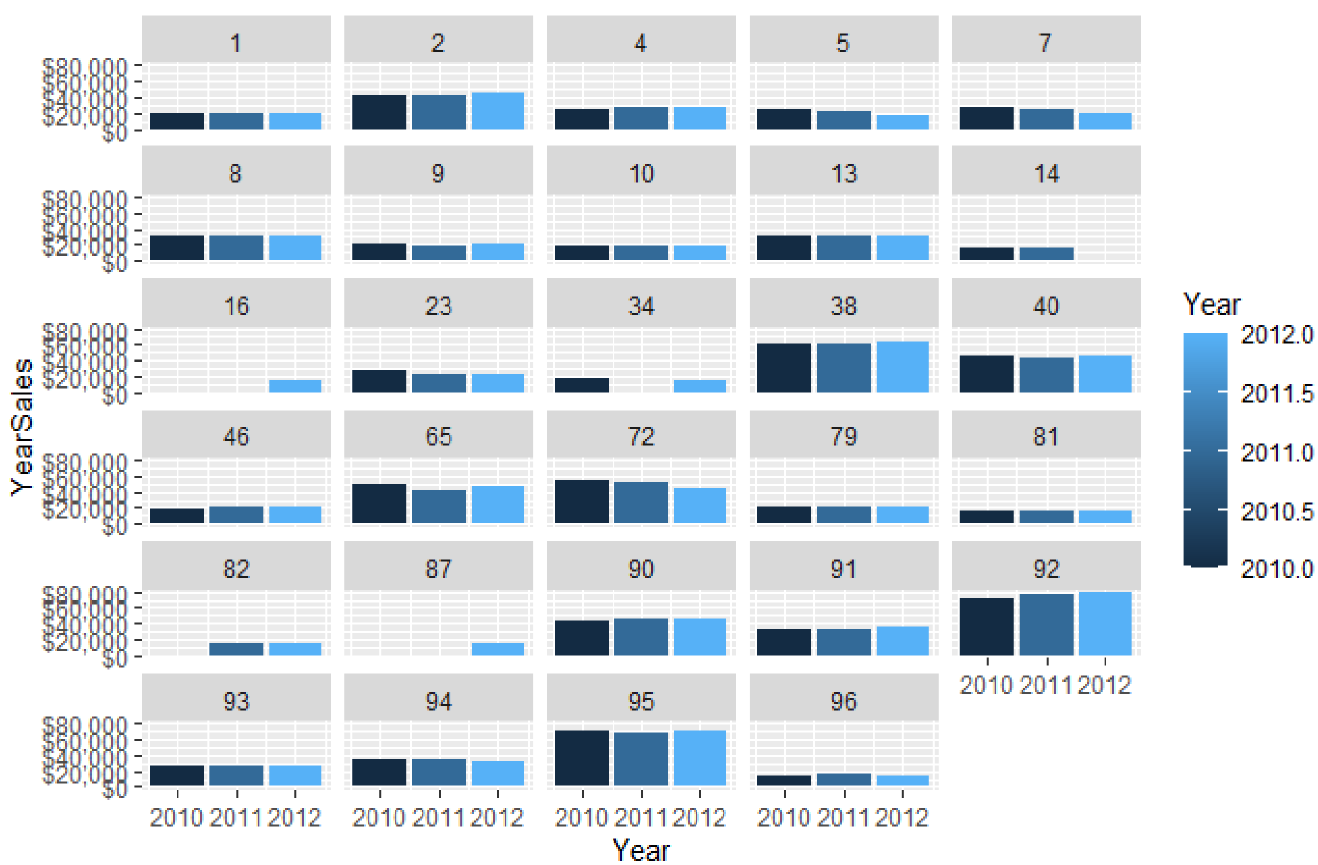


- Plot of Sales Distribution by Department

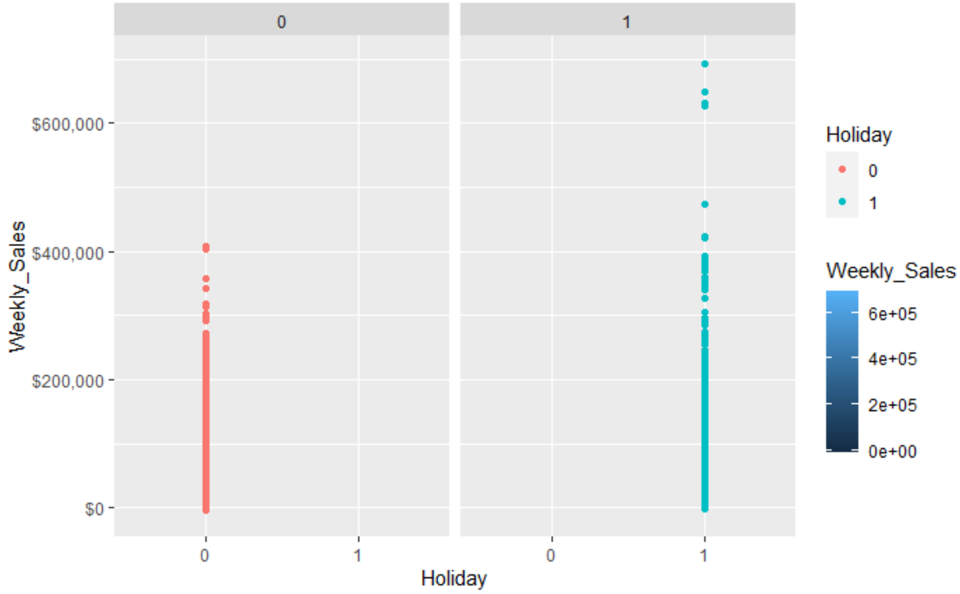


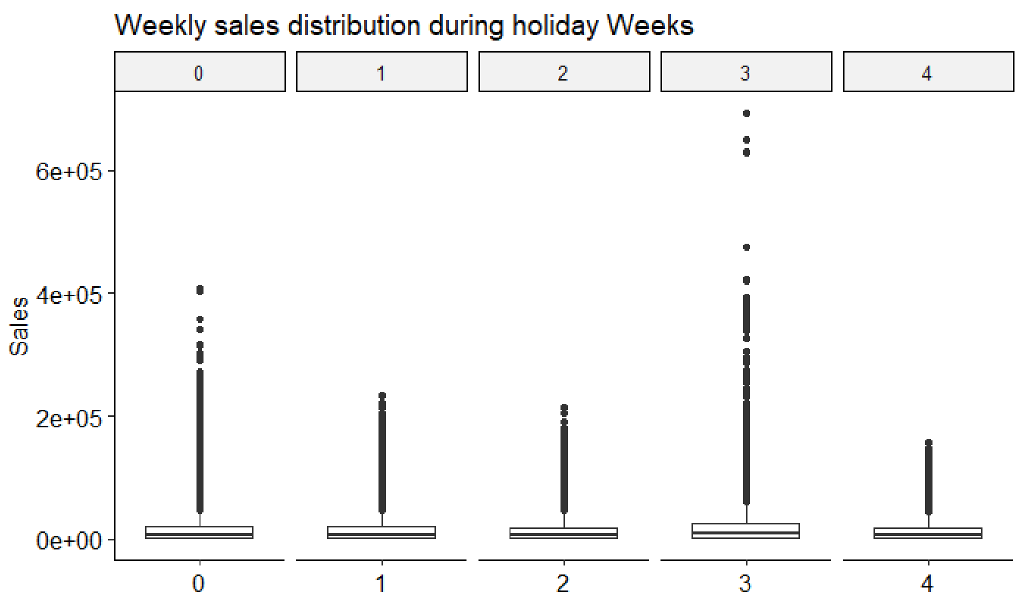


Average weekly sales at each Department per year



- Bivariate Analysis – “Weekly Sales” vs “Holiday”





Holiday Week	0	Super Bowl Week (1)	Labour Day Week (2)	Thanksgiving & Black Friday Week (3)	Christmas Week (4)
Number of Records	391909	8895	8861	5959	5946
Sum of Weekly Sales	6231919436	145682278	140727685	132414609	86474980
Percentage of Total Weekly Sales	92.49%	2.16%	2.08%	1.96%	1.28%
Avg Weekly Sales	15901.45	16378.00	15881.69	22220.94	14543.39

- Significant sales are happening on holidays. We can identify the days on which the holidays occur from the given dataset as below –

```
 # Filter holiday set
hdf <- df1 %>% filter(df1$Holiday==1)
ulst <- unique(hdf$Date2)
ulst
```

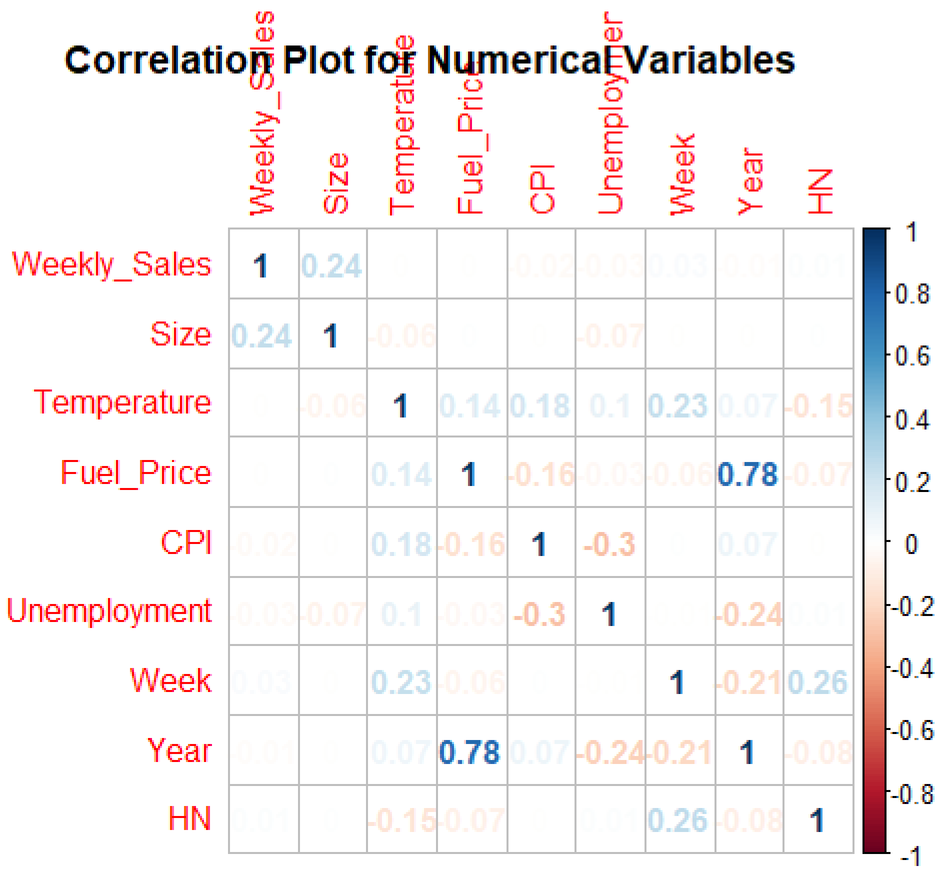
"2012-02-10" "2012-09-07"
"2011-02-11" "2011-09-09" "2011-11-25" "2011-12-30"
"2010-02-12" "2010-09-10" "2010-11-26" "2010-12-31"

- Super Bowl: 12-Feb-2010, 11-Feb-2011, 10-Feb-2012 **(first Sunday in February)**
- Labor Day: 10-Sep-2010, 9-Sep-2011, 7-Sep-2012 **(first Monday of September)**
- Thanksgiving: 26-Nov-2010, 25-Nov-2011 **(fourth Thursday of November)**
- Christmas: 31-Dec-2010, 30-Dec-2011 **(typical holidays are from 25th December to 5th January)**

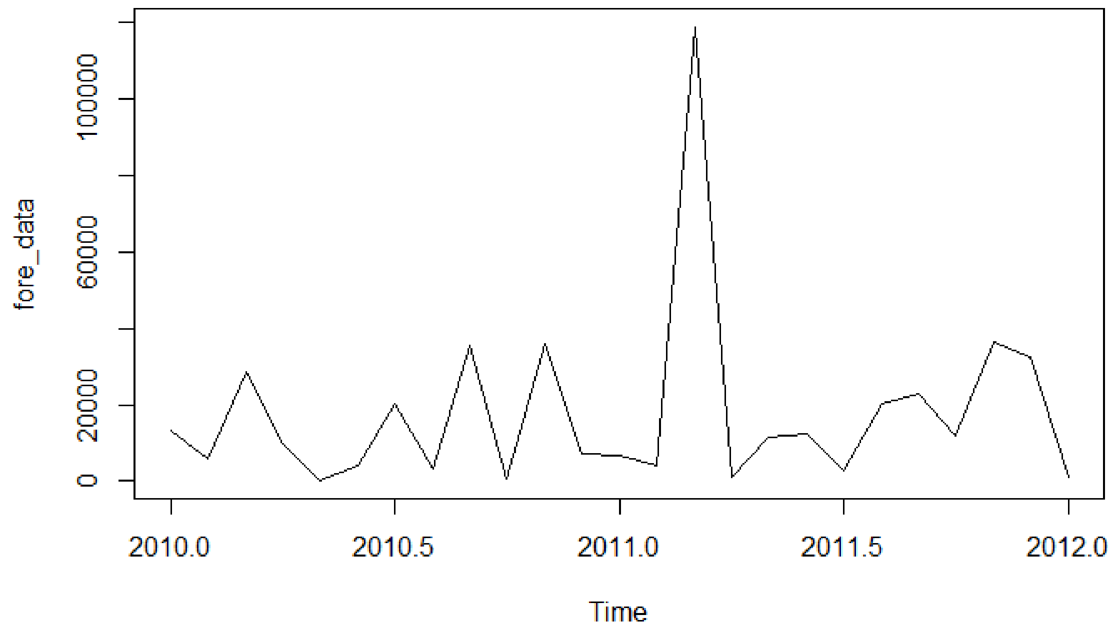
Thanksgiving Day is the day with the highest food consumption of the year. It is followed by Black Friday which follows after Thanksgiving Day (which is usually the fourth Friday of November) marks the beginning of the Christmas shopping season and has become the busiest shopping day and the day with the highest retail turnover of the year in the United States.

Analysis of sales during holiday weeks can help stores keep inventory as well as staff as per demand.

- Used the library(corrplot) to plot correlation and check for high correlation between the numerical variables. Correlation plot does not show high correlation between the variables.



- Use timeseries package to plot Weekly Sales vs Year. Unusual high sales is shown in early year 2011.



NORMALIZE AND SPLIT DATASET

Sometimes we need to normalize data in order to compare different variables that are not in the same scale. In this case CPI, Unemployment rate, Temperature, Markdowns and sales are all different levels. If we don't normalize these variables the weight in some predictive models could be very different.

The function to normalize data is (x - min(x))/(max(x) - min(x)).

We take only the numerical values to normalize. For this sake we duplicate dataset and exclude factor variables to numerical type.

To ease the modelling performance, we split the main dataset to 4 Subsets based on the grouping of Departments ->
Subset 1 -> Type A Stores
Subset 2-> Type B Stores
Subset 3 -> Type C Stores

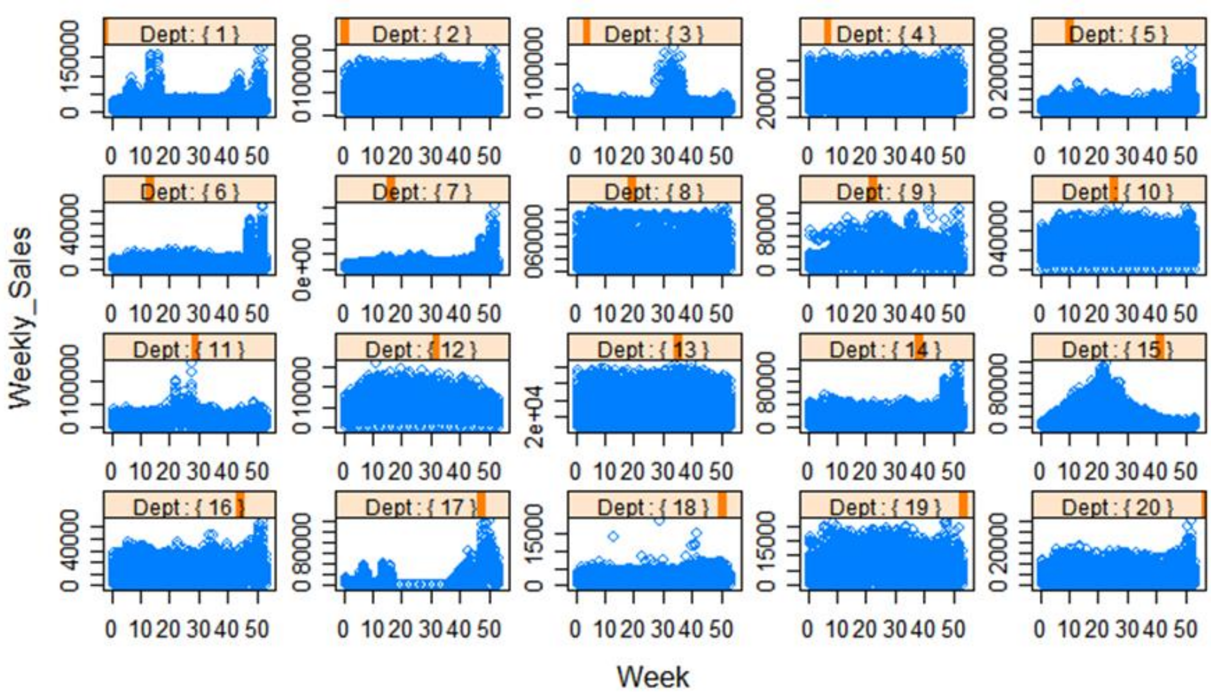
Above is the split for train and test data to perform predictive modelling using ensemble learning techniques such as Gradient Boosting Models, Random Forest, and Decision Tree. The results are compared against Liner Regression models to select best performing model for prediction.

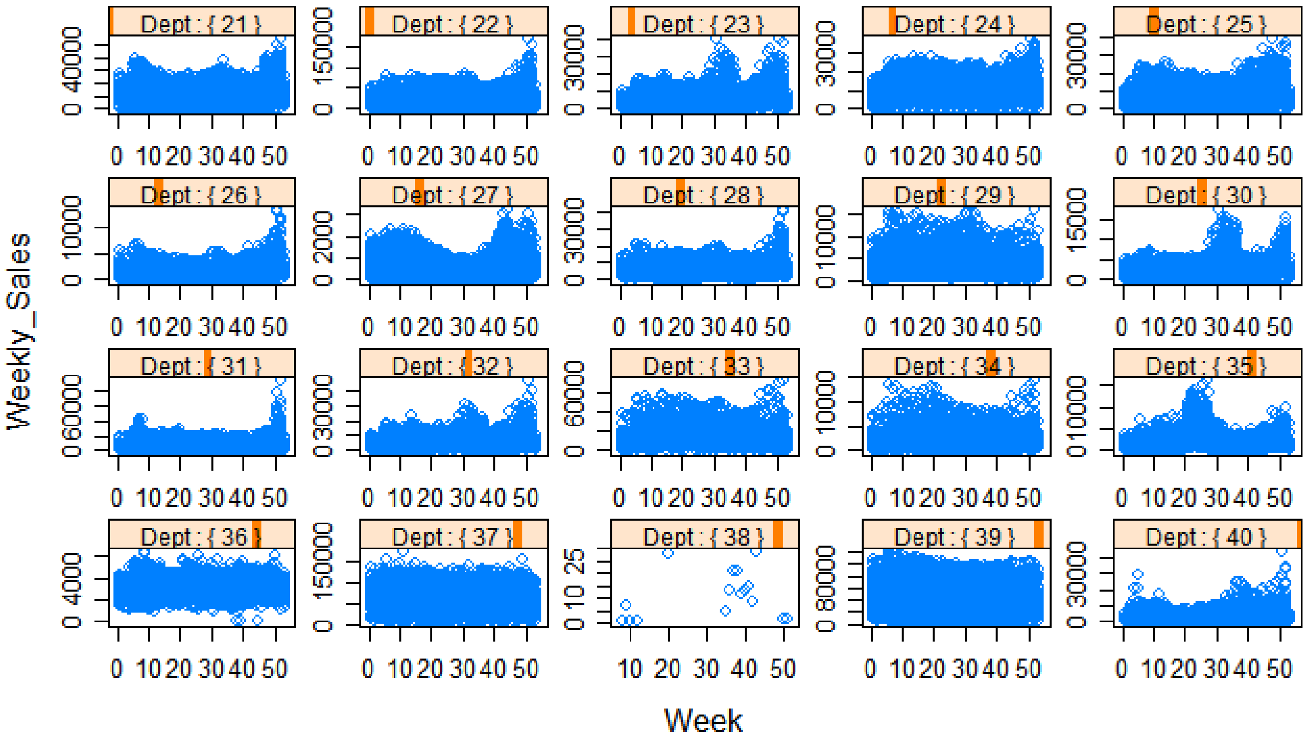
Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.

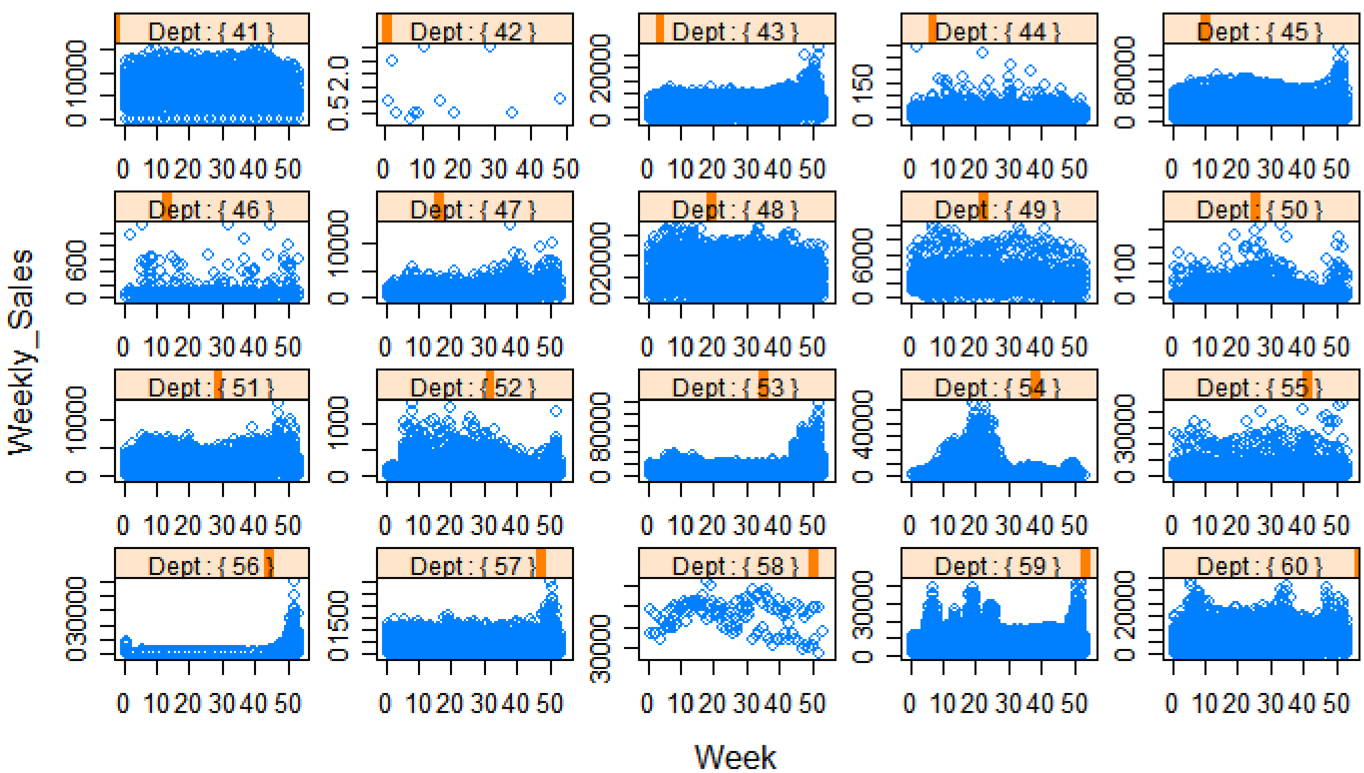
Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Whereas random forests build an ensemble of deep independent trees, GBMs uses Boosting framework to build an ensemble of weak successive trees . that iteratively improves with each tree learning and improving on the previous.

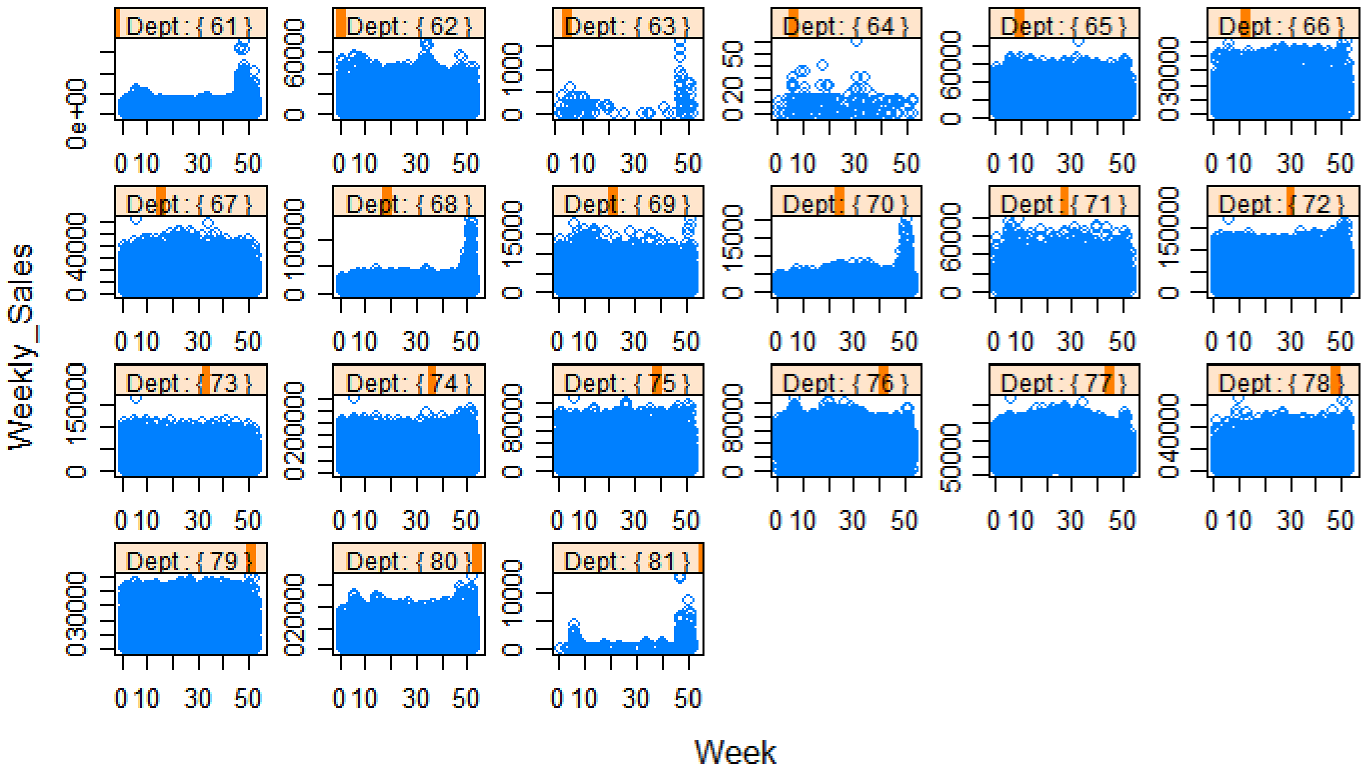
Here is cumulative weekly sale s distribution graph per Department -




Dept : { 21 }
Dept : { 22 }
Dept : { 23 }
Dept : { 24 }
Dept : { 25 }
Dept : { 26 }
Dept : { 27 }
Dept : { 28 }
Dept : { 29 }
Dept : { 30 }
Dept : { 31 }
Dept : { 32 }
Dept : { 33 }
Dept : { 34 }
Dept : { 35 }
Dept : { 36 }
Dept : { 37 }
Dept : { 38 }
Dept : { 39 }
Dept : { 40 }
Weekly_Sales
Week


Dept : { 41 }
Dept : { 42 }
Dept : { 43 }
Dept : { 44 }
Dept : { 45 }
Dept : { 46 }
Dept : { 47 }
Dept : { 48 }
Dept : { 49 }
Dept : { 50 }
Dept : { 51 }
Dept : { 52 }
Dept : { 53 }
Dept : { 54 }
Dept : { 55 }
Dept : { 56 }
Dept : { 57 }
Dept : { 58 }
Dept : { 59 }
Dept : { 60 }
Weekly_Sales
Week

[redacted]


Dept : { 61 }
Dept : { 62 }
Dept : { 63 }
Dept : { 64 }
Dept : { 65 }
Dept : { 66 }
Dept : { 67 }
Dept : { 68 }
Dept : { 69 }
Dept : { 70 }
Dept : { 71 }
Dept : { 72 }
Dept : { 73 }
Dept : { 74 }
Dept : { 75 }
Dept : { 76 }
Dept : { 77 }
Dept : { 78 }
Dept : { 79 }
Dept : { 80 }
Dept : { 81 }
Weekly_Sales
Week

GBM MODELS

MODEL 1 (ON SUBSET 1)

> summary(fit.gbm1)

gbm(formula = Weekly_Sales ~ Week + Month + Year + Size + CPI + Unemployment + Temperature, distribution = "gaussian", data = df.train.1, n.trees = GBM_Ntrees, interaction.depth = 10, shrinkage = GBM_Shrinkage, bag.fraction = GBM_Bag.fraction, cv.folds = 16, verbose = F, n.cores = 2)

A gradient boosted model with gaussian loss function.

500 iterations were performed.

The best cross-validation iteration was 452.

There were 7 predictors of which 7 had non-zero influence.

With External Variables -

	var <chr>	rel.inf <dbl>
Size	Size	72.12008767
Week	Week	14.48425387
Unemployment	Unemployment	6.50989996
CPI	CPI	5.85176676
Temperature	Temperature	0.62213083
Year	Year	0.31757263
Month	Month	0.09428829

7 rows

With Internal Variables –

gbm(formula = Weekly_Sales ~ Week + Month + Year + Size + MarkDown1 + MarkDown2 + MarkDown3 + MarkDown4 + MarkDown5, distribution = "gaussian", data = df.train.1, n.trees = GBM_Ntrees, interaction.depth = 10, shrinkage = GBM_Shrinkage, bag.fraction = GBM_Bag.fraction, cv.folds = 16, verbose = F, n.cores = 2)

A gradient boosted model with gaussian loss function.

500 iterations were performed.

The best cross-validation iteration was 480.

There were 9 predictors of which 9 had non-zero influence.

	var <chr>	rel.inf <dbl>
Size	Size	83.73479531
Week	Week	14.27637986
MarkDown4	MarkDown4	0.53950407
MarkDown1	MarkDown1	0.48727389
MarkDown5	MarkDown5	0.43832397
Year	Year	0.17735483
MarkDown3	MarkDown3	0.15948254
Month	Month	0.10816016
MarkDown2	MarkDown2	0.07872538

9 rows

Confusion Matrix and RMSE

[1] "Accuracy :- 81.3799540025561"
[1] "FNR :- 18.6200459974439"
[1] "FPR :- 18.6200459974439"
[1] "precision :- 81.3799540025561"
[1] "recall//TPR :- 81.3799540025561"
[1] "Sensitivity :- 81.3799540025561"
[1] "Specificity :- 81.3799540025561"

sqrt(min(fit.gbm1$cv.error))	Train RMSE	Test RMSE
18926.43	18873.12	18897.12

MODEL 2 (ON SUBSET 2)

gbm(formula = Weekly_Sales ~ Week + Month + Year + Size + CPI + Unemployment + Temperature, distribution = "gaussian", data = df.train.2, n.trees = GBM_Ntrees, interaction.depth = 10, shrinkage = GBM_Shrinkage, bag.fraction = GBM_Bag.fraction, cv.folds = 16, verbose = F, n.cores = 2)

A gradient boosted model with gaussian loss function.

500 iterations were performed.

The best cross-validation iteration was 389.

There were 7 predictors of which 7 had non-zero influence.

With External Variables -

	var <chr>	rel.inf <dbl>
Size	Size	39.04781929
CPI	CPI	37.98053599
Week	Week	11.81350622
Unemployment	Unemployment	10.57494943
Temperature	Temperature	0.37410520
Year	Year	0.16939551
Month	Month	0.03968837

7 rows

With Internal Variables –

gbm(formula = Weekly_Sales ~ Week + Month + Year + Size + CPI + Unemployment + Temperature, distribution = "gaussian", data = df.train.2, n.trees = GBM_Ntrees, interaction.depth = 10, shrinkage = GBM_Shrinkage, bag.fraction = GBM_Bag.fraction, cv.folds = 16, verbose = F, n.cores = 2)

A gradient boosted model with gaussian loss function.

500 iterations were performed.

The best cross-validation iteration was 389.

There were 7 predictors of which 7 had non-zero influence.

	var <chr>	rel.inf <dbl>
Size	Size	87.32019656
Week	Week	11.49430332
MarkDown1	MarkDown1	0.27866601
MarkDown4	MarkDown4	0.22615029
MarkDown5	MarkDown5	0.19727734
Year	Year	0.18699311
MarkDown3	MarkDown3	0.17196187
Month	Month	0.07422548
MarkDown2	MarkDown2	0.05022602

Confusion Matrix and RMSE

[1] "Accuracy :- 75.9877536849342"
[1] "FNR :- 24.0122463150658"
[1] "FPR :- 24.0122463150658"
[1] "precision :- 75.9877536849342"
[1] "recall//TPR :- 75.9877536849342"
[1] "Sensitivity :- 75.9877536849342"
[1] "Specificity :- 75.9877536849342"

sqrt(min(fit.gbm2$cv.error))	Train RMSE	Test RMSE
14212.84	14152.51	14174.01

MODEL 3 (ON SUBSET 3)

gbm(formula = Weekly_Sales ~ Week + Month + Year + Size + CPI + Unemployment + Temperature, distribution = "gaussian", data = df.train.3, n.trees = GBM_Ntrees, interaction.depth = 10, shrinkage = GBM_Shrinkage, bag.fraction = GBM_Bag.fraction, cv.folds = 16, verbose = F, n.cores = 2)

A gradient boosted model with gaussian loss function.

500 iterations were performed.

The best cross-validation iteration was 90.

There were 7 predictors of which 5 had non-zero influence.

With External Variables -

	var <chr>	rel.inf <dbl>
Week	Week	43.1575393
Unemployment	Unemployment	24.9220819
CPI	CPI	17.6375885
Size	Size	10.0054440
Temperature	Temperature	3.0171909
Year	Year	0.8134889
Month	Month	0.4466665

With Internal Variables –

gbm(formula = Weekly_Sales ~ Week + Month + Year + Size + MarkDown1 + MarkDown2 + MarkDown3 + MarkDown4 + MarkDown5, distribution = "gaussian", data = df.train.3, n.trees = GBM_Ntrees, interaction.depth = 10, shrinkage = GBM_Shrinkage, bag.fraction = GBM_Bag.fraction, cv.folds = 16, verbose = F, n.cores = 2)

A gradient boosted model with gaussian loss function.

500 iterations were performed.

The best cross-validation iteration was 46.

There were 9 predictors of which 8 had non-zero influence.

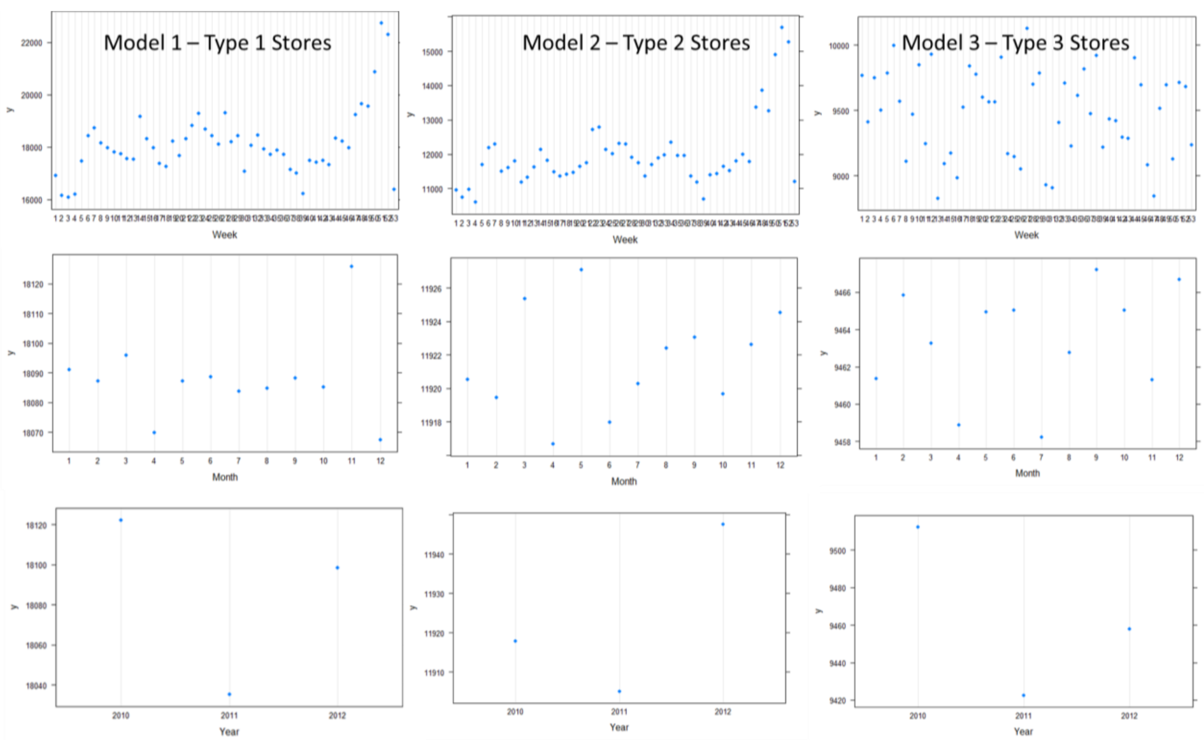
	var <chr>	rel.inf <dbl>
Week	Week	53.4109963
Size	Size	26.1013983
MarkDown5	MarkDown5	6.1676533
MarkDown1	MarkDown1	5.6143990
MarkDown3	MarkDown3	2.8802517
MarkDown4	MarkDown4	2.3330369
Year	Year	2.2149476
Month	Month	0.6898233
MarkDown2	MarkDown2	0.5874936

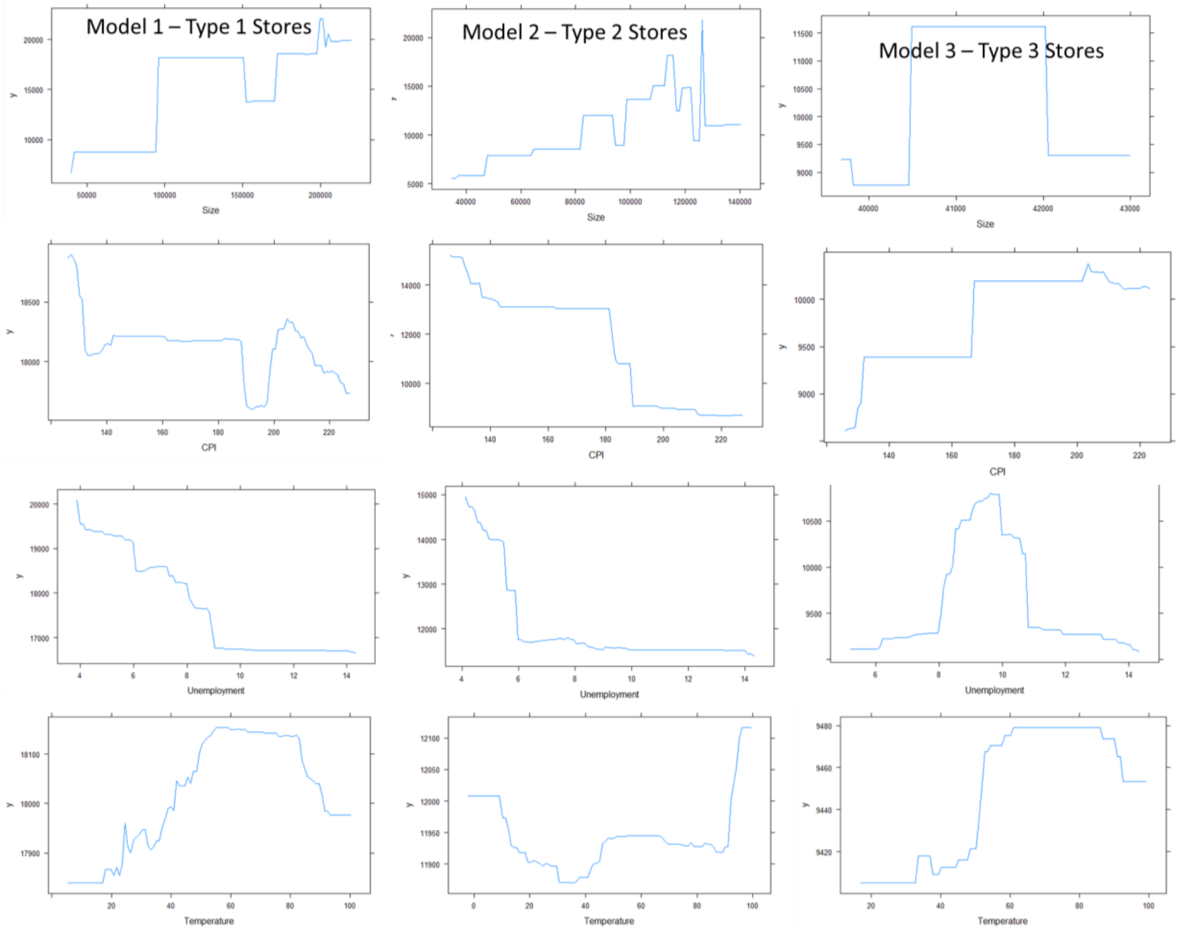
9 rows

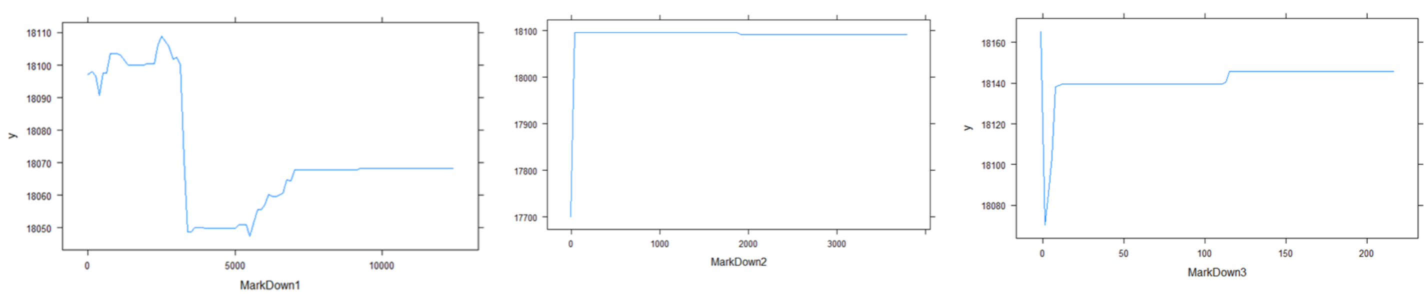
Confusion Matrix and RMSE

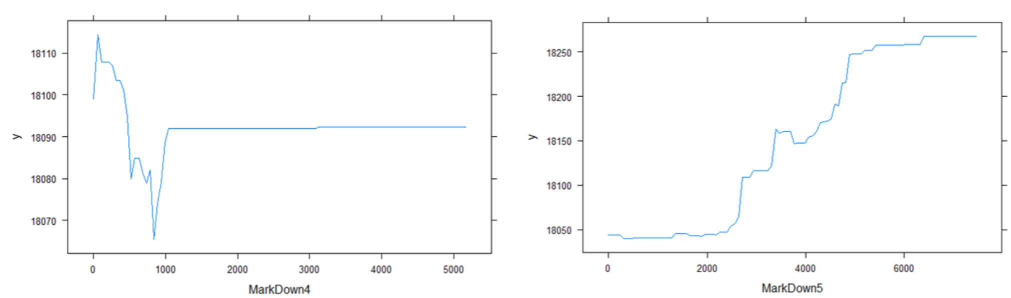
[1] "Accuracy :- 89.6531575669645"
[1] "FNR :- 10.3468424330355"
[1] "FPR :- 10.3468424330355"
[1] "precision :- 89.6531575669645"
[1] "recall//TPR :- 89.6531575669645"
[1] "Sensitivity :- 89.6531575669645"
[1] "Specificity :- 89.6531575669645"

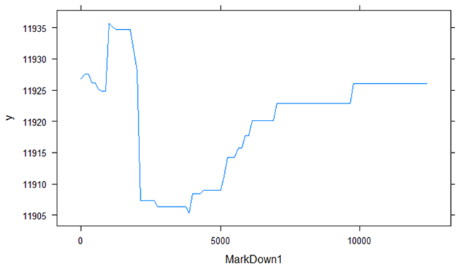
sqrt(min(fit.gbm3$cv.error))	Train RMSE	Test RMSE
15430.54	15352.1	15325.19

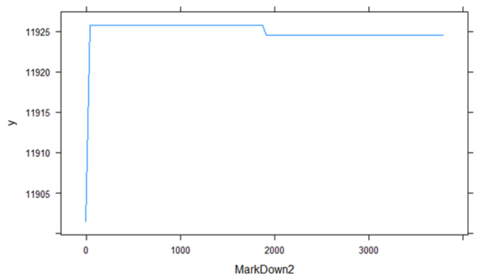


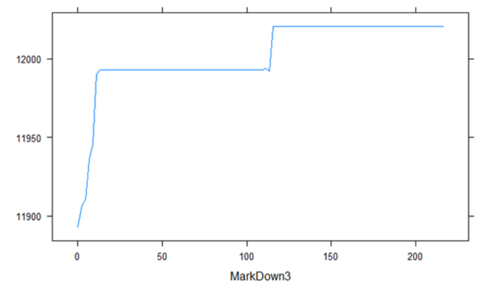

Model 1 – Type 1 Stores
Model 2 – Type 2 Stores
Model 3 – Type 3 Stores
Size
Size
Size
CPI
CPI
CPI
Unemployment
Unemployment
Unemployment
Temperature
Temperature
Temperature

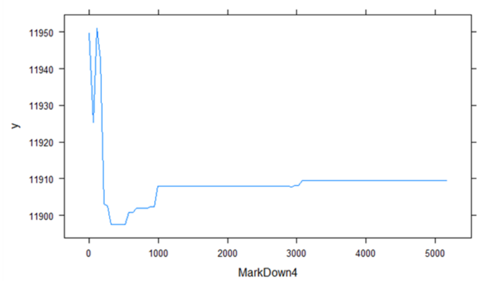

MarkDown1
MarkDown2
MarkDown3

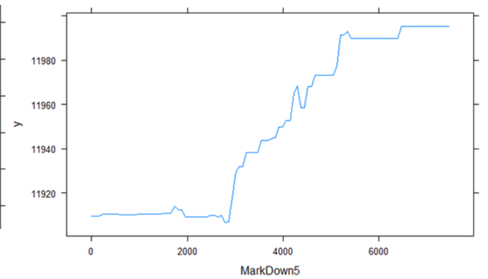

MarkDown4
MarkDown5

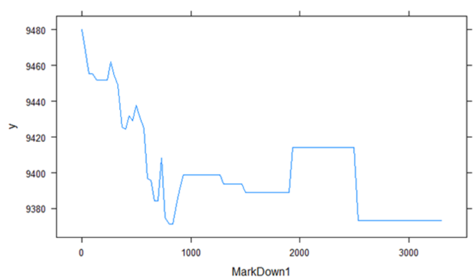

y
MarkDown1

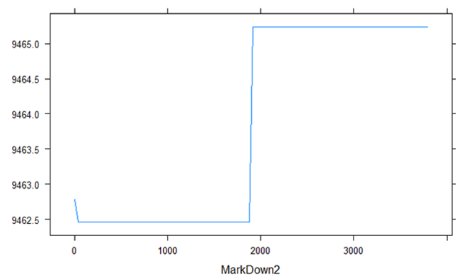

y
MarkDown2

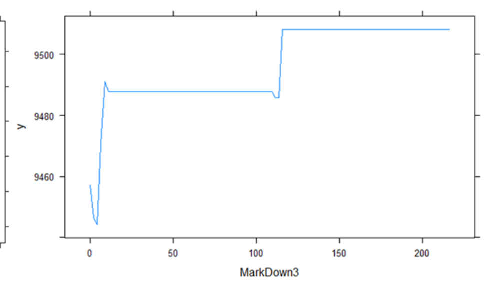

y
MarkDown3

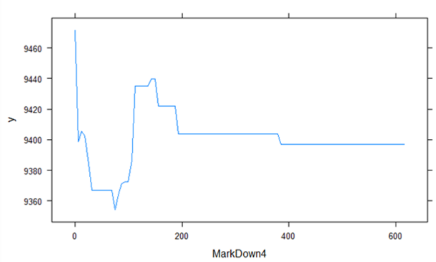

y
MarkDown4

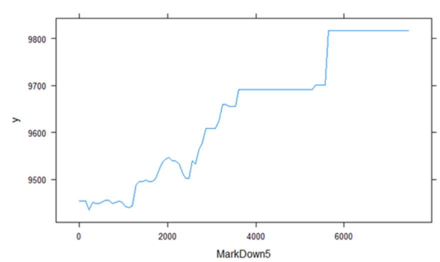

y
MarkDown5


y
MarkDown1


MarkDown2


y
MarkDown3

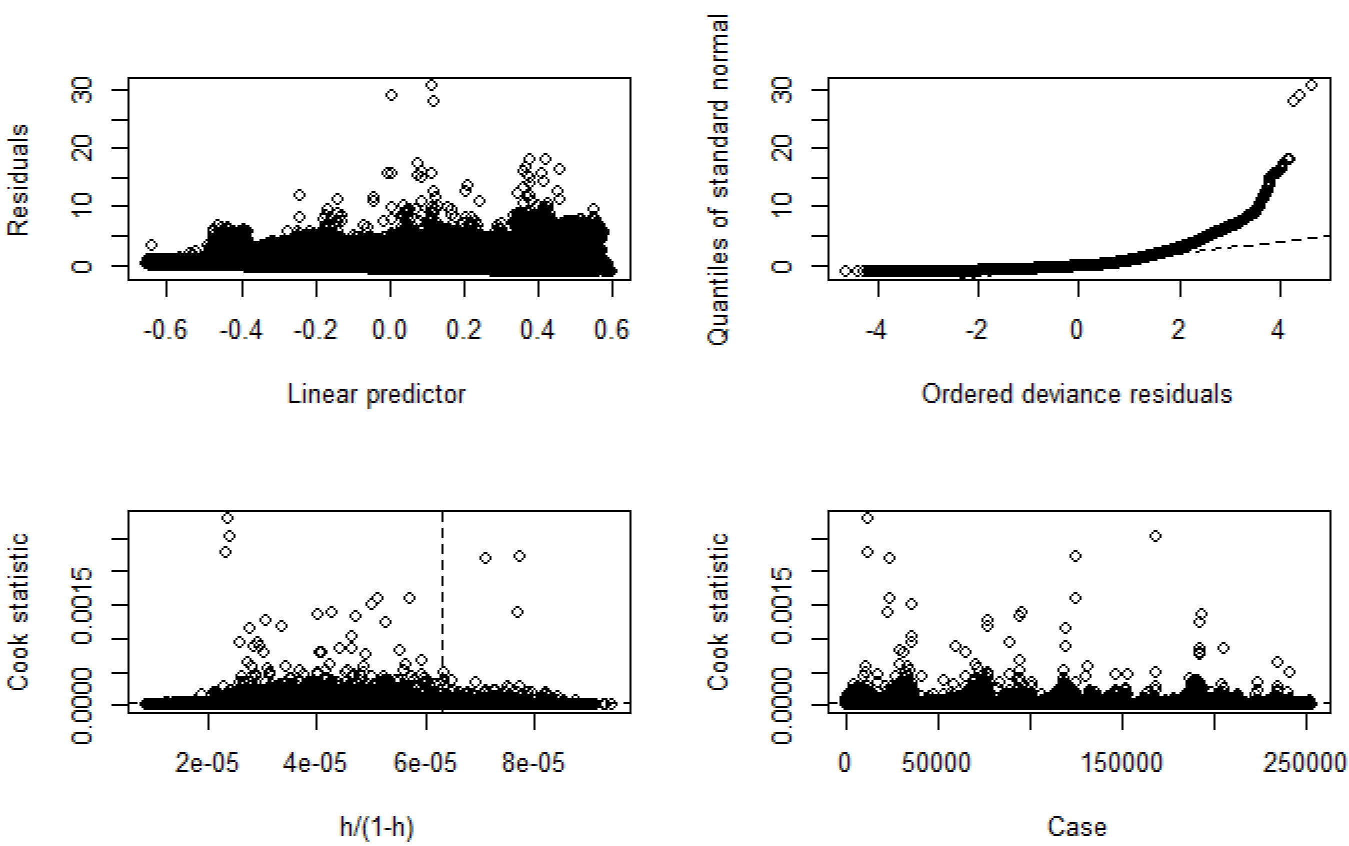

y
MarkDown4


y
MarkDown5

LINEAR REGRESSION MODEL

MODEL 1 (ON ENTIRE DATASET)

```
Call:
lm(formula = Weekly_Sales ~ Dept + Week + Size + CPI + Temperature +
    Fuel_Price + Unemployment, data = df.train)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3025 -0.5759 -0.2590  0.2355 29.6984

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.0002436  0.0019150   0.127 0.898784
Dept          0.1129493  0.0019161  58.948  < 2e-16 ***
Week          0.0234916  0.0019855  11.832  < 2e-16 ***
Size          0.2434112  0.0019211 126.702  < 2e-16 ***
CPI          -0.0275753  0.0021208 -13.002  < 2e-16 ***
Temperature   0.0133622  0.0020876   6.401 1.55e-10 ***
Fuel_Price   -0.0065867  0.0020012  -3.291 0.000997 ***
Unemployment -0.0194879  0.0020596  -9.462  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9631 on 252935 degrees of freedom
Multiple R-squared:  0.07411,	Adjusted R-squared:  0.07409
F-statistic:  2892 on 7 and 252935 DF,  p-value: < 2.2e-16
```



> summary(aov(model1))

```
                  Df Sum Sq Mean Sq   F value    Pr(>F)    
Dept               1   3318    3318  3577.204   < 2e-16 ***
Week               1    187     187   201.930   < 2e-16 ***
Size               1  15086   15086 16263.243   < 2e-16 ***
CPI                1     84      84    90.330   < 2e-16 ***
Temperature        1     18      18    19.229 0.0000116 ***
Fuel_Price         1      4       4     4.392    0.0361 *  
Unemployment       1     83      83    89.529   < 2e-16 ***
Residuals     252935 234626       1                        
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Important Variables

> car::vif(model1)

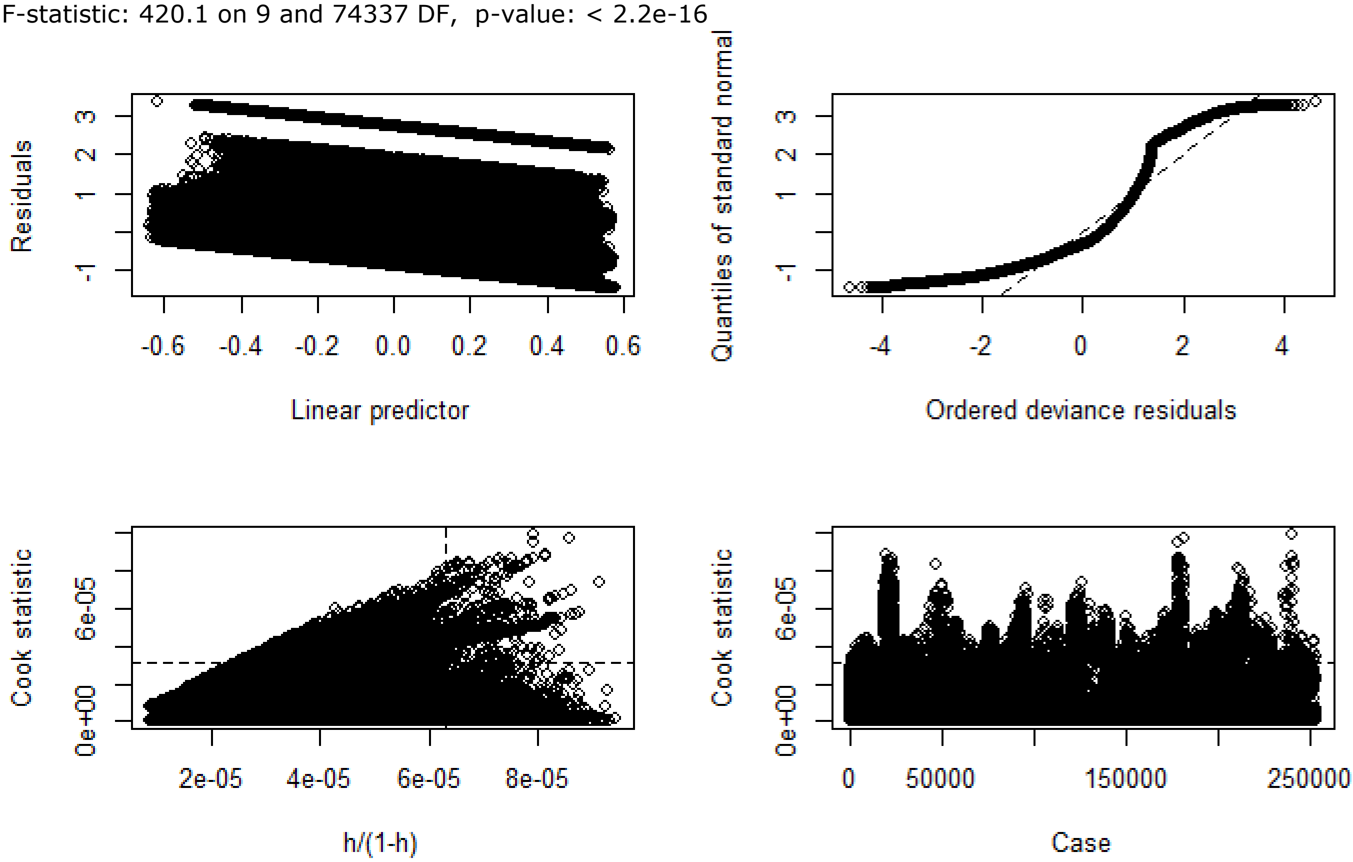
```
    Dept         Week         Size          CPI  Temperature   Fuel_Price Unemployment 
1.000150     1.075971     1.007992     1.226880     1.187257     1.092695     1.155606 
```

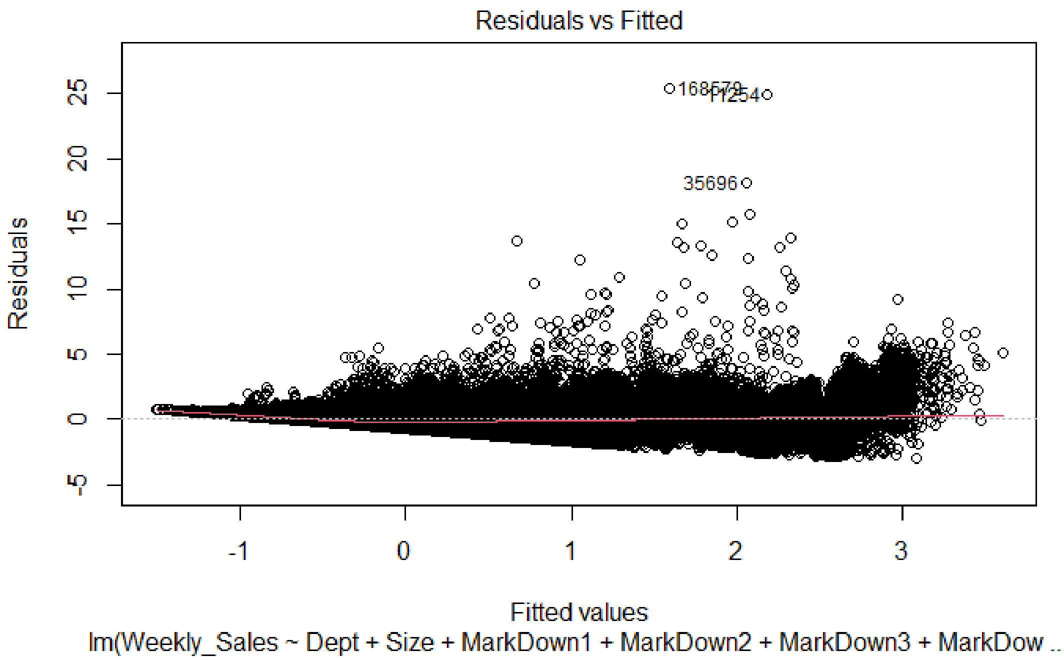
Confusion Matrix

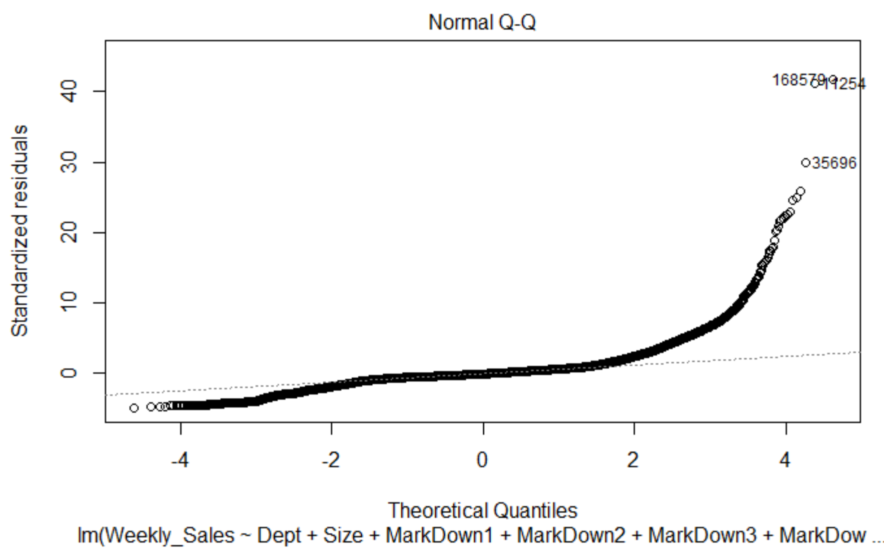
```
[1] "Accuracy :- 78.7130560761136"
[1] "FNR :- 21.2869439238864"
[1] "FPR :- 21.2869439238864"
[1] "precision :-  78.7130560761136"
[1] "recall//TPR :-  78.7130560761136"
[1] "Sensitivity :-  78.7130560761136"
[1] "Specificity :-  78.7130560761136"
```

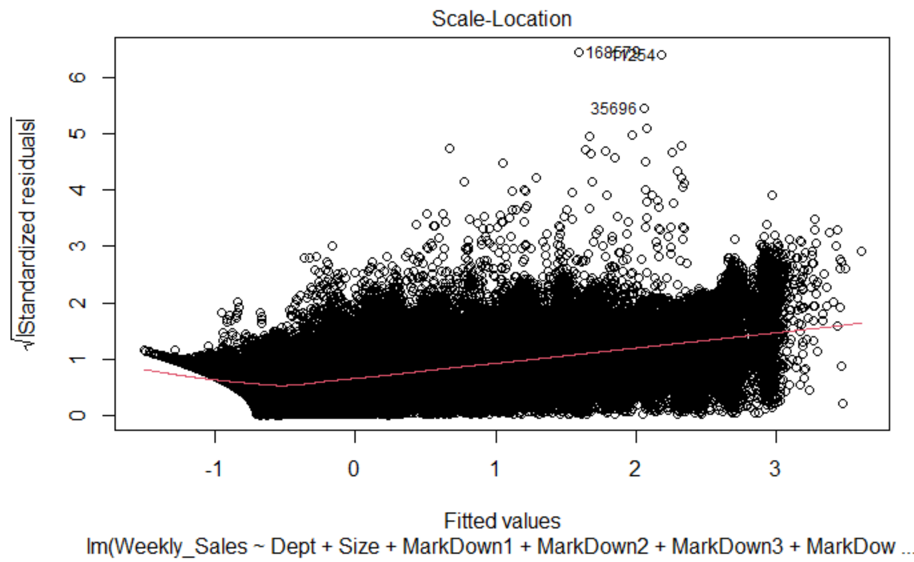
MODEL 2 (ON ENTIRE DATASET)

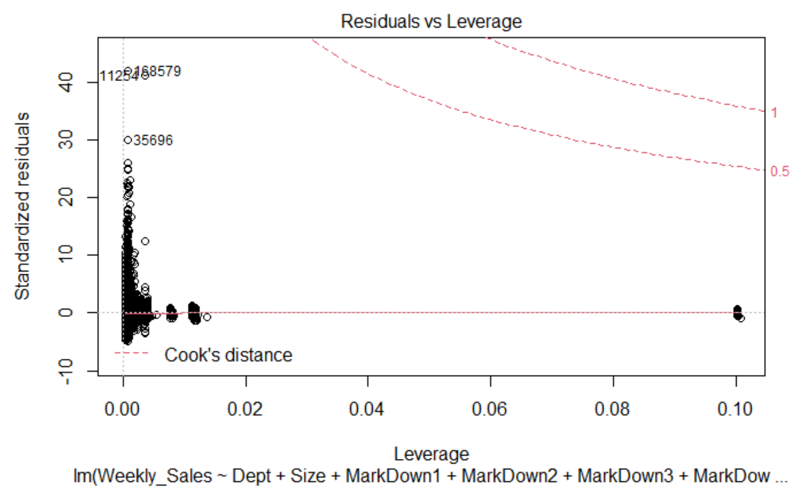
```
Call:
lm(formula = Weekly_Sales ~ Dept + Size + MarkDown1 + MarkDown2 +
    MarkDown3 + MarkDown4 + MarkDown5 + Week + HN, data = norm.df.train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.07693 -0.03277 -0.01636  0.01226  0.94308 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.0110624  0.0006468  17.102  < 2e-16 ***
Dept         0.0115194  0.0006660  17.297  < 2e-16 ***
Size         0.0362075  0.0006392  56.645  < 2e-16 ***
MarkDown1   -0.0002806  0.0046433  -0.060   0.9518    
MarkDown2    0.0026303  0.0036418   0.722   0.4701    
MarkDown3    0.0220085  0.0043571   5.051 4.40e-07 ***
MarkDown4    0.0032425  0.0047843   0.678   0.4979    
MarkDown5    0.0062906  0.0024633   2.554   0.0107 *  
Week         0.0058850  0.0007752   7.592 3.19e-14 ***
HN          -0.0001885  0.0014692  -0.128   0.8979    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05437 on 74337 degrees of freedom
Multiple R-squared:  0.0484,	Adjusted R-squared:  0.04829 
F-statistic: 420.1 on 9 and 74337 DF,  p-value: < 2.2e-16
```




Residuals vs Fitted
Residuals
Fitted values
lm(Weekly_Sales ~ Dept + Size + MarkDown1 + MarkDown2 + MarkDown3 + MarkDow ...
168579
11254
35696


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Weekly_Sales ~ Dept + Size + MarkDown1 + MarkDown2 + MarkDown3 + MarkDow ...
168579
11254
35696





> summary(aov(model2))

```
> summary(aov(model2))
               Df Sum Sq Mean Sq  F value   Pr(>F)
Dept            1   1.01   1.011  342.095  < 2e-16 ***
Size            1   9.80   9.804 3315.897  < 2e-16 ***
MarkDown1       1   0.01   0.007    2.483 0.115096
MarkDown2       1   0.00   0.003    0.853 0.355779
MarkDown3       1   0.13   0.132   44.768 2.23e-11 ***
MarkDown4       1   0.00   0.000    0.000 0.999386
MarkDown5       1   0.04   0.041   13.910 0.000192 ***
Week            1   0.18   0.181   61.224 5.16e-15 ***
HN              1   0.00   0.000    0.016 0.897939
Residuals   74337 219.78   0.003
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

```
> car::vif(model2)
    Dept     Size MarkDown1 MarkDown2 MarkDown3 MarkDown4 MarkDown5      Week        HN 
1.000461 1.015948  4.225349  1.262325  1.171873  3.615584  1.450025  1.132165  1.442163 
```

Confusion Matrix

```
> calc(cm.lm2)
[1] "Accuracy :- 78.6988650137505"
[1] "FNR :- 21.3011349862495"
[1] "FPR :- 21.3011349862495"
[1] "precision :-  78.6988650137505"
[1] "recall//TPR :-  78.6988650137505"
[1] "Sensitivity :-  78.6988650137505"
[1] "Specificity :-  78.6988650137505"
```

DECISION TREES

We use rpart – Recursive Partitioning and Regression Trees for classification. rpart uses K-fold cross validation to validate the optimal complexity parameter (cp). The model also plots important variables as given below.

Following are the subset data for initial modelling purposes.

df.train1 <- df.subset1[index,]

df.test1 <- df.subset1[-index,]

Using the important parameters derived from the above, the rpart is performed on the entire dataset

df.train <- df[index,]

df.test <- df[-index,]

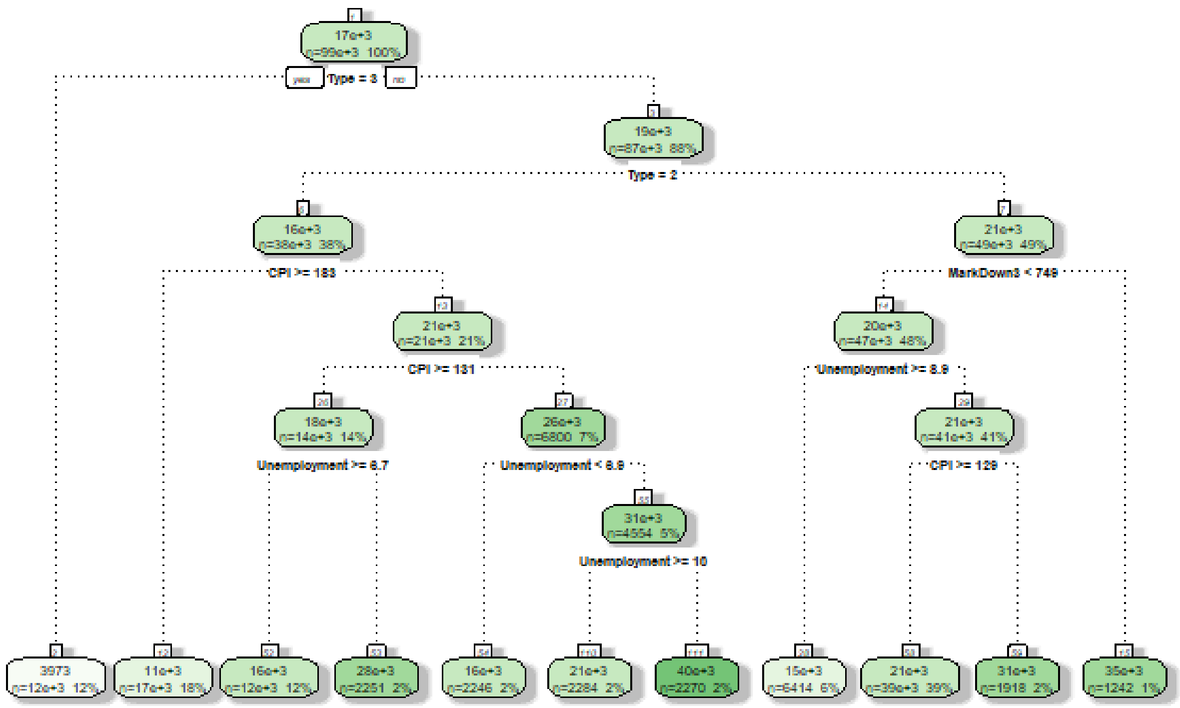
MODEL 1 (ON SUBSET 1)

Call:
rpart(formula = Weekly_Sales ~ CPI + Unemployment + Temperature + MarkDown3 + Type + HN, data = df.train1[, -c(15)], control = r.ctrl)
n= 99127

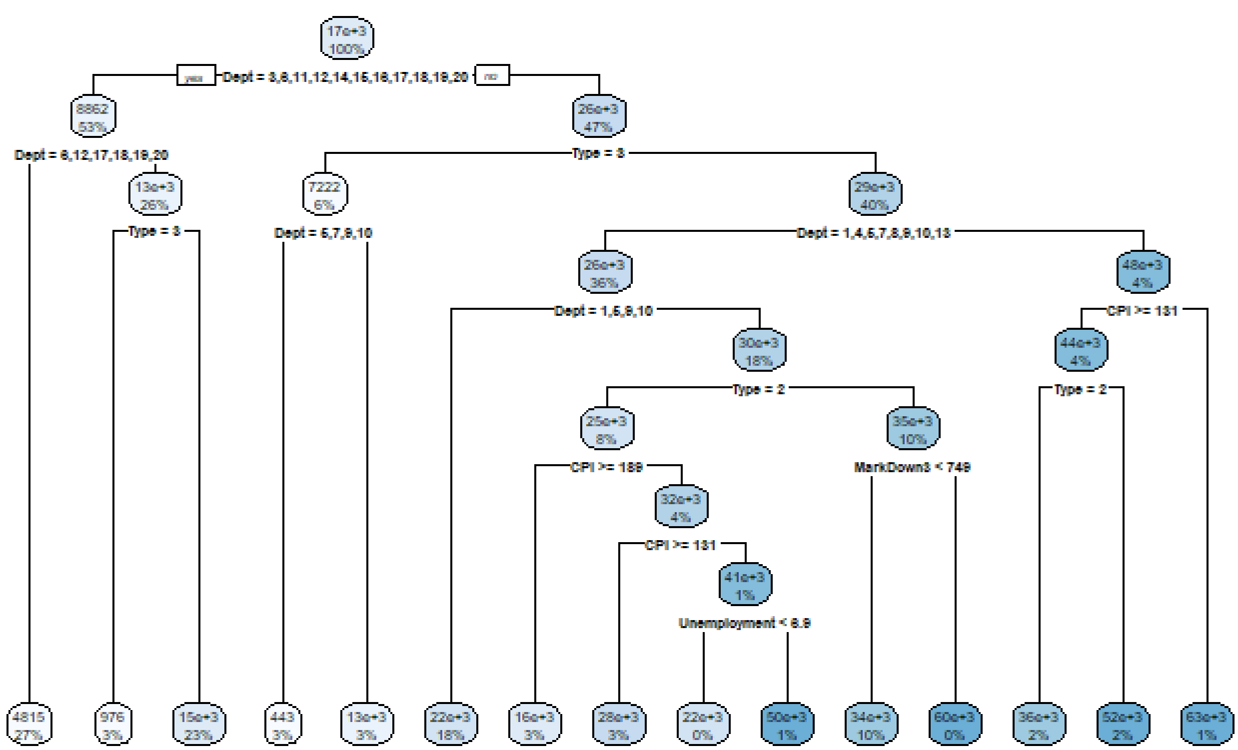
	CP	nsplit	rel error	xerror	xstd
1	0.071489558	0	1.0000000	1.0000180	0.01303429
2	0.018773567	1	0.9285104	0.9285319	0.01267714
3	0.010468895	3	0.8909633	0.8895702	0.01230387
4	0.007716610	6	0.8595566	0.8585745	0.01181162
5	0.007522926	7	0.8518400	0.8554767	0.01170325
6	0.006704325	8	0.8443171	0.8479527	0.01155879
7	0.005697351	9	0.8376128	0.8380999	0.01145940
8	0.005000000	10	0.8319154	0.8324073	0.01134583

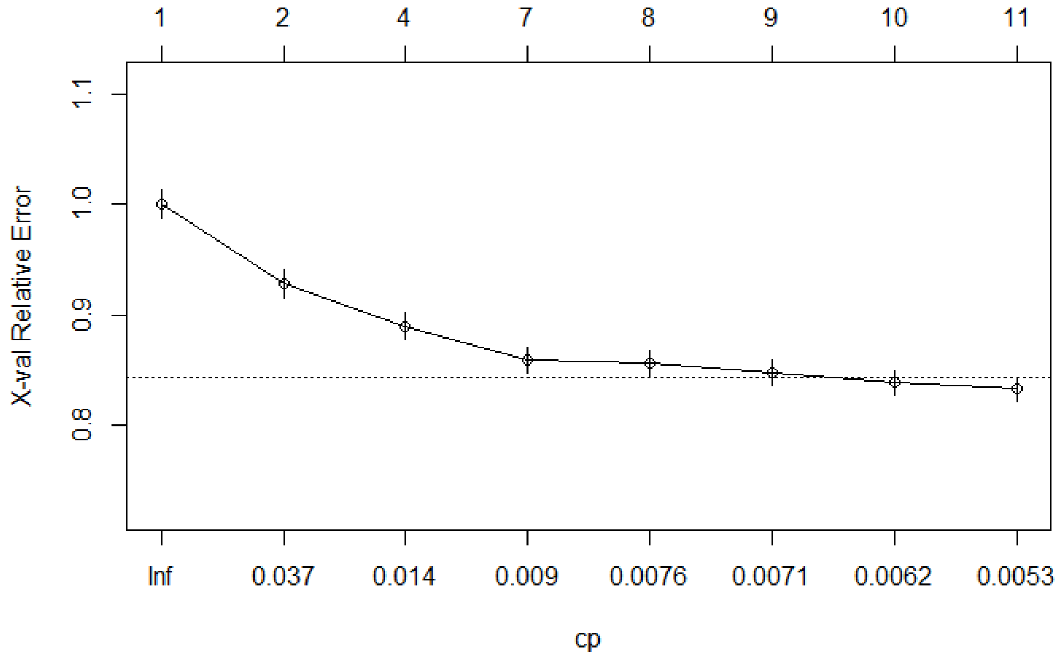
Variable Importance

```
Variable importance
        Type Unemployment          CPI  Temperature    MarkDown3 
          43           26           21            5            4 
```



Rattle 2020-Sep-15 16:41:54 Shilpa





MODEL 2 (ON SUBSET 2)

rpart(formula = Weekly_Sales ~ MarkDown1 + MarkDown2 + MarkDown3 +

MarkDown4 + MarkDown5 + Type + HN, data = df.train1[, -c(15)],

control = r.ctrl)

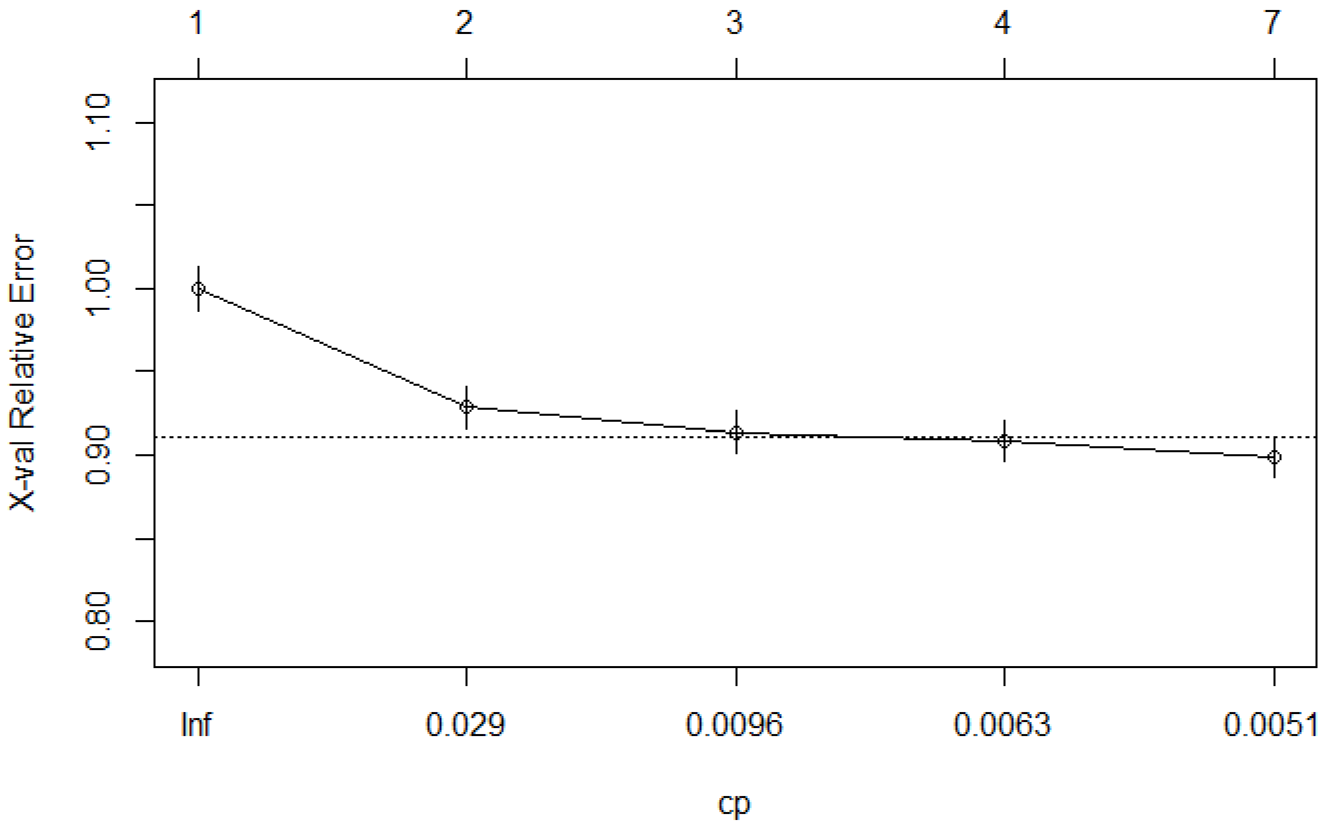
Variables actually used in tree construction:

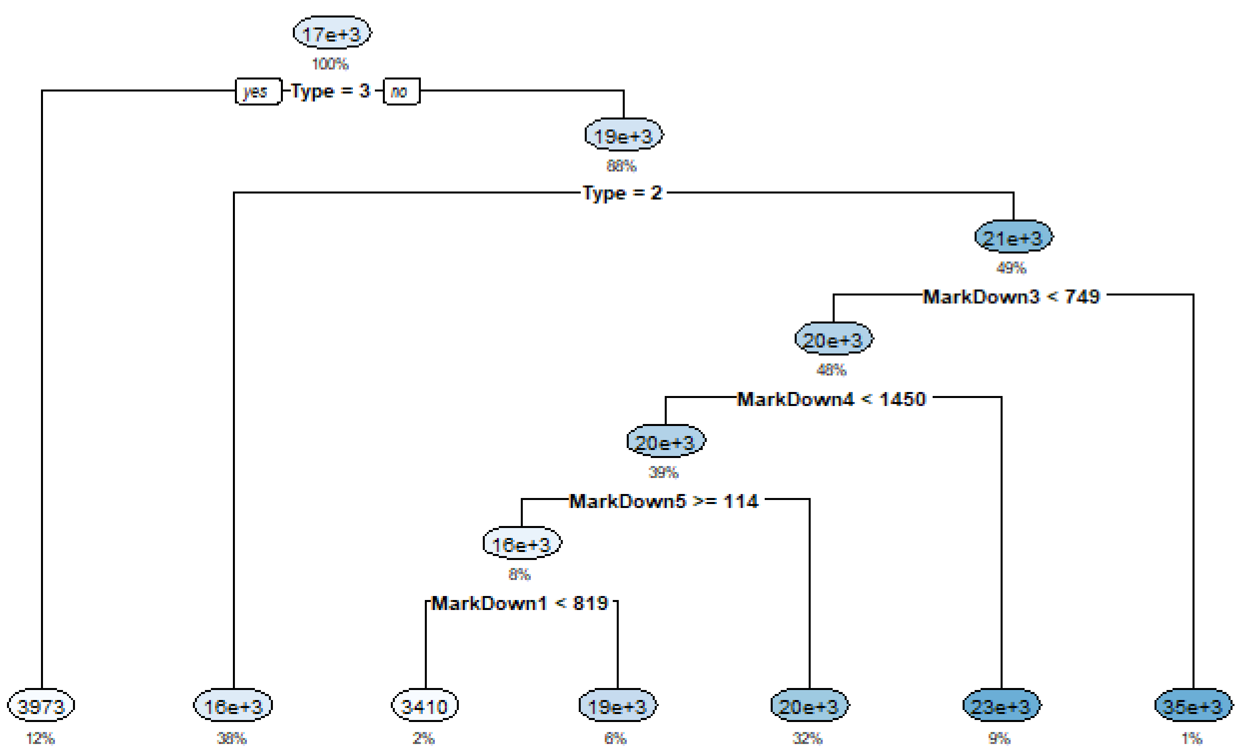
[1] MarkDown1 MarkDown3 MarkDown4 MarkDown5 Type

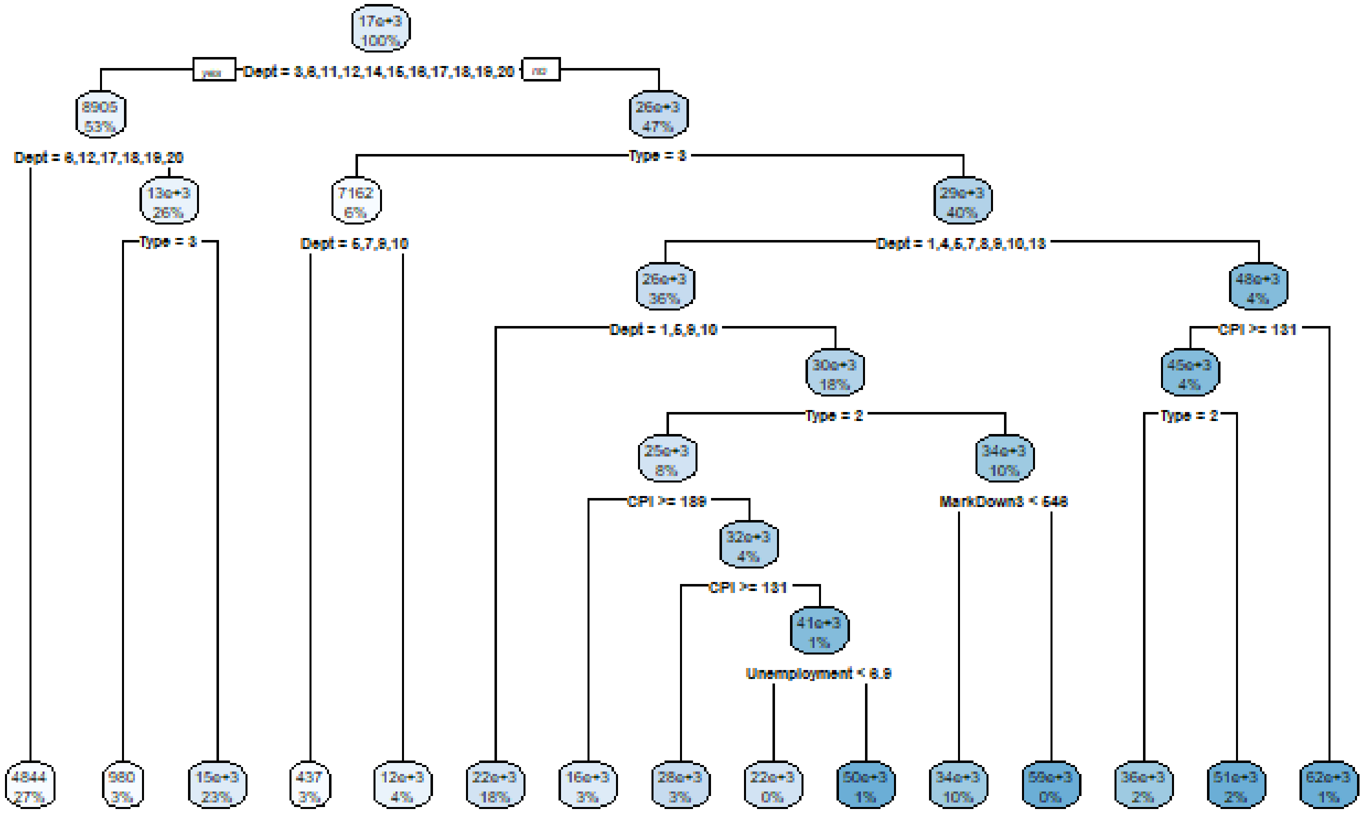
Root node error: 3.246e+13/99127 = 327462863

n= 99127

```
Variable importance
     Type MarkDown1 MarkDown3 MarkDown4 MarkDown5 MarkDown2
       61        11        10         9         7         2
```


1
2
3
4
7
X-val Relative Error
1.10
1.00
0.90
0.80
Inf
0.029
0.0096
0.0063
0.0051
cp


17e+3
100%
yes
Type = 3
no
19e+3
88%
Type = 2
21e+3
49%
MarkDown3 < 749
20e+3
48%
MarkDown4 < 1450
20e+3
39%
MarkDown5 >= 114
16e+3
8%
MarkDown1 < 819
3973
12%
16e+3
38%
3410
2%
19e+3
6%
20e+3
32%
23e+3
9%
35e+3
1%

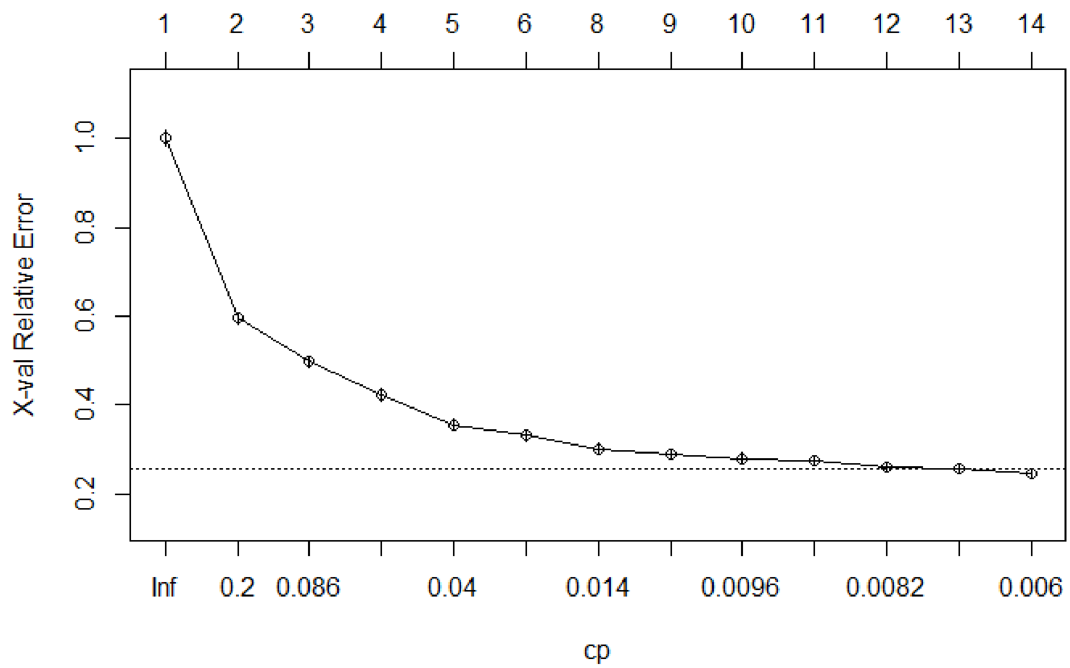


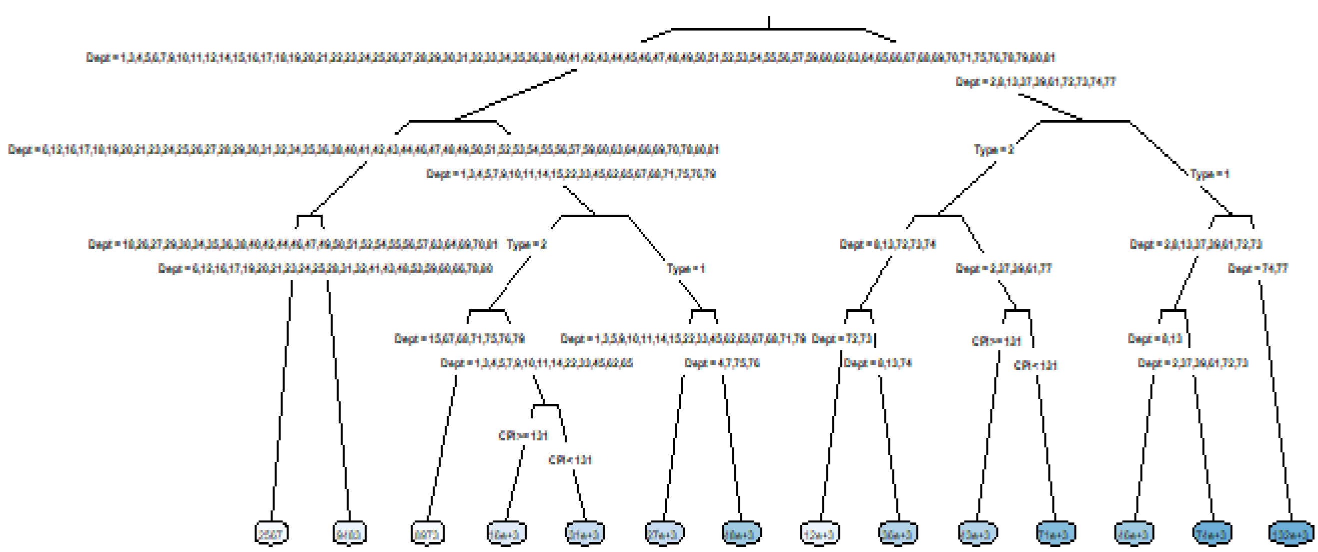
17e+3
100%
yes
Dept = 3,6,11,12,14,15,16,17,18,19,20
no
8905
53%
26e+3
47%
Dept = 6,12,17,18,19,20
Type = 3
13e+3
26%
7162
6%
29e+3
40%
Type = 3
Dept = 5,7,8,10
Dept = 1,4,5,7,8,9,10,13
26e+3
36%
48e+3
4%
Dept = 1,5,9,10
CPI >= 131
30e+3
18%
45e+3
4%
Type = 2
Type = 2
25e+3
8%
34e+3
10%
CPI >= 189
MarkDown3 < 548
32e+3
4%
CPI >= 131
41e+3
1%
Unemployment < 8.9
4844
27%
980
3%
15e+3
23%
437
3%
12e+3
4%
22e+3
18%
16e+3
3%
28e+3
3%
22e+3
0%
50e+3
1%
34e+3
10%
59e+3
0%
36e+3
2%
51e+3
2%
62e+3
1%

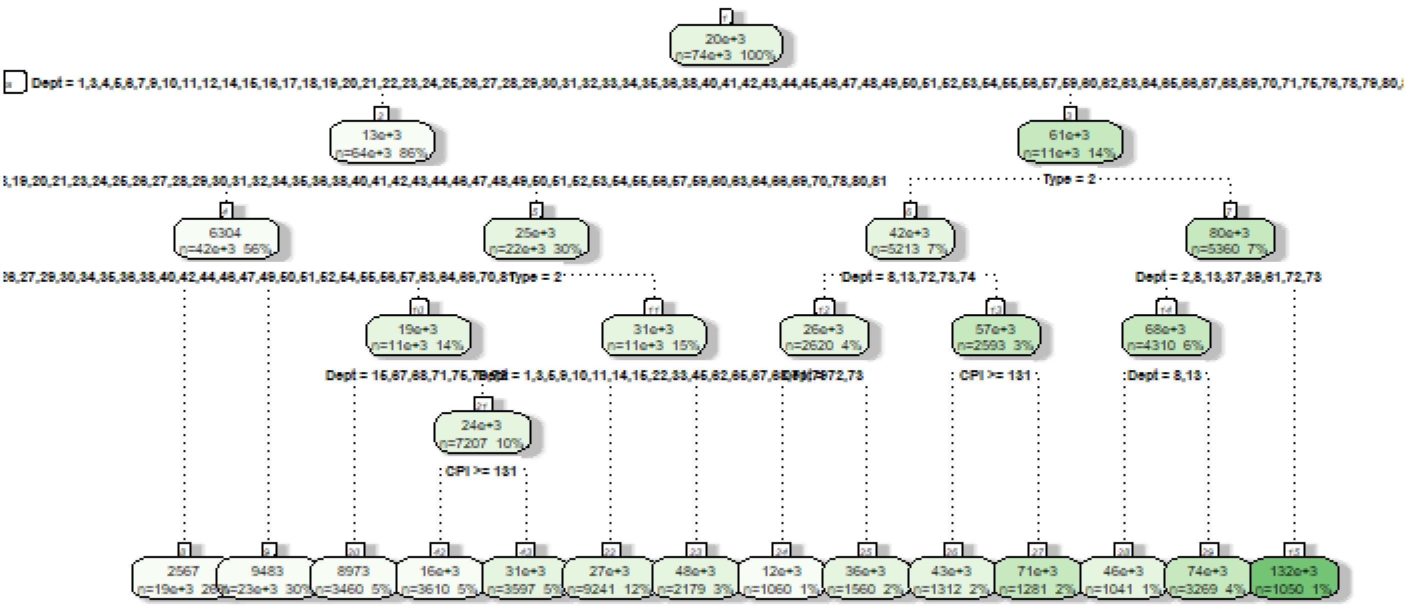
MODEL 3 (ON ENTIRE DATASET)

Regression tree:
rpart(formula = Weekly_Sales ~ Dept + CPI + Unemployment + Temperature + MarkDown3 + MarkDown5 + Type + HN, data = df.train[, -c(15)], control = r.ctrl)

Variables actually used in tree construction:
[1] CPI Dept Type
Root node error: 5.2386e+13/74347 = 704610237
n= 74347







Rattle 2020-Sep-15 18:42:59 Shilpa

Prediction

rpart.prediction <- predict(train.rpart,df.test, type="vector")

```
> calc(cm.rpart)
[1] "Accuracy :- 88.3480825958702"
[1] "FNR :- 100"
[1] "FPR :- 7.98771121351766"
[1] "precision :-  0"
[1] "recall//TPR :-  0"
[1] "Sensitivity :-  0"
[1] "Specificity :-  92.0122887864823"
```

Regression tree:

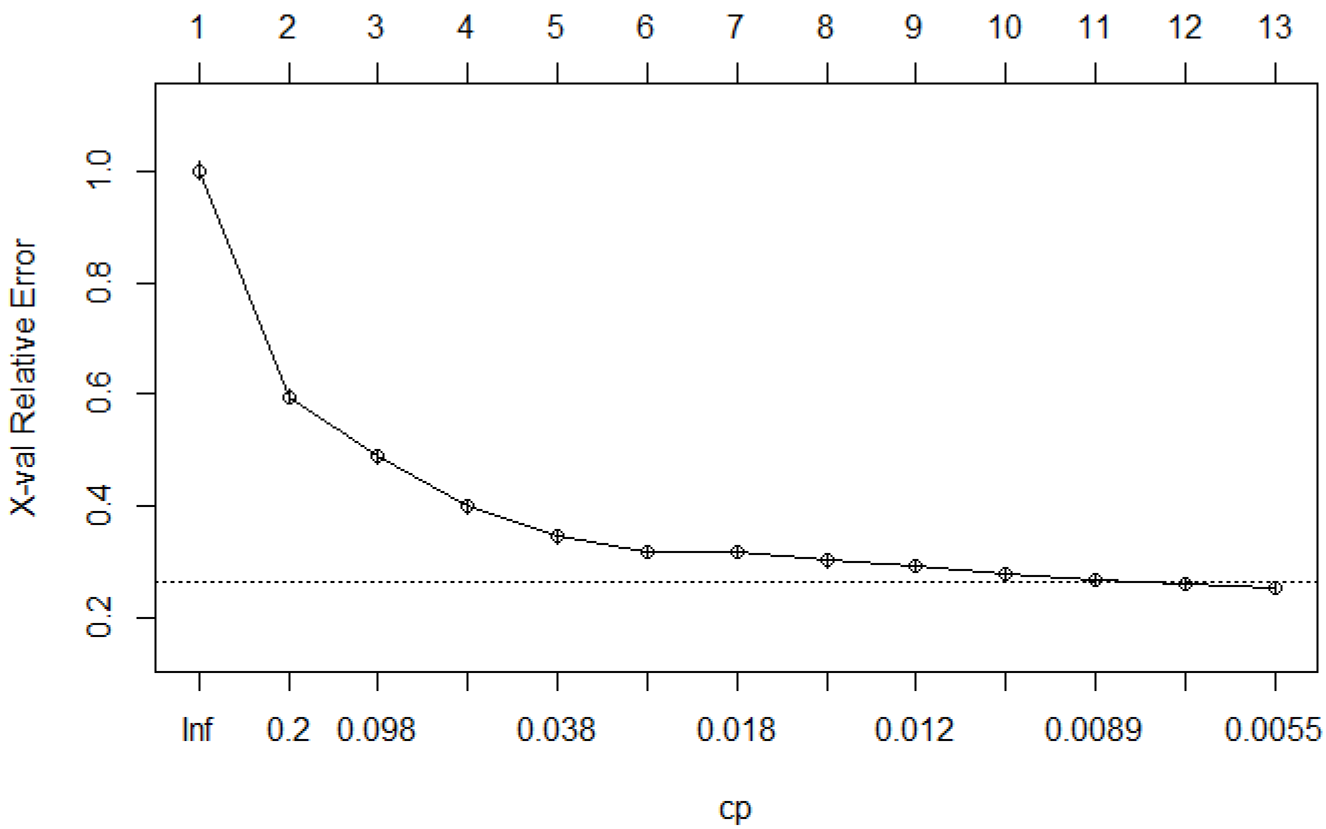
rpart(formula = Weekly_Sales ~ Dept + Store + MarkDown1 + MarkDown2 + MarkDown3 + MarkDown4 + MarkDown5 + Month + Type, data = df.train[,-c(15)], control = r.ctrl)

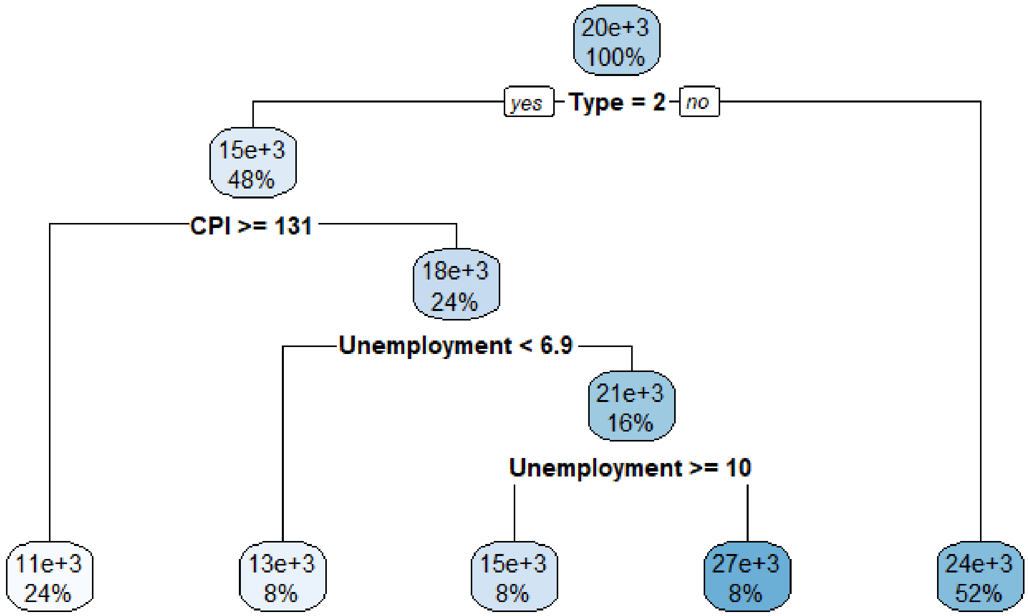
Variables actually used in tree construction:

[1] Dept Store

Root node error: 5.2386e+13/74347 = 704610237

n= 74347





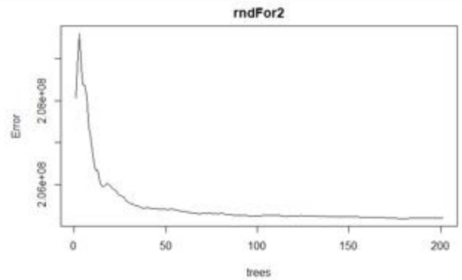

```
> calc(cm.rpart)
[1] "Accuracy :- 91.0133843212237"
[1] "FNR :- 100"
[1] "FPR :- 8.81226053639847"
[1] "precision :-  0"
[1] "recall//TPR :-  0"
[1] "Sensitivity :-  0"
[1] "Specificity :-  91.1877394636015"
```

RANDOM FOREST

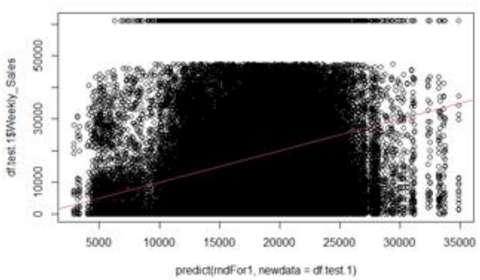
MODEL 1

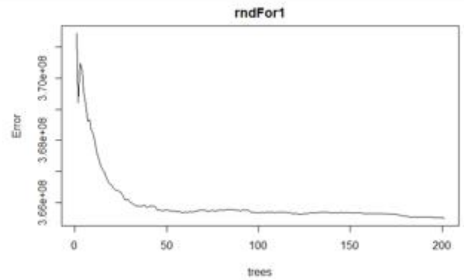
Call:
randomForest(formula = Weekly_Sales ~ ., data = df.train[, -15], ntree = 101, mtry = 3, nodesize = 10, importance = TRUE)
Type of random forest: regression
Number of trees: 101
No. of variables tried at each split: 3
Mean of squared residuals: 289284733
% Var explained: 58.94

```
> impVar[order(impVar[,2],decreasing = TRUE),]
             %IncMSE IncNodePurity
Dept           41.17  2.012610e+13
Size           16.47  2.616257e+12
Type            9.26  1.353362e+12
Store          12.71  1.281371e+12
CPI            14.14  1.137865e+12
Unemployment   10.38  7.531497e+11
Temperature     7.65  5.918914e+11
Week            8.15  5.472732e+11
Fuel_Price      5.90  5.372920e+11
Month           6.84  2.665928e+11
MarkDown3       1.78  2.442191e+11
MarkDown5       1.77  2.194203e+11
MarkDown1       2.16  2.107912e+11
MarkDown4       4.01  1.992302e+11
MarkDown2       2.59  1.613994e+11
HN              0.39  1.424665e+11
Year            6.92  5.907415e+10
```

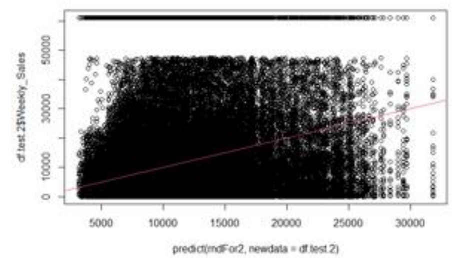


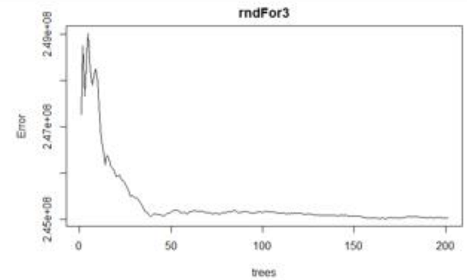
	%IncMSE	IncNodePurity
Size	63.36	1.618260e+12
Week	-9.94	7.473634e+11
CPI	25.47	4.950382e+11
Unemployment	22.59	4.777259e+11
Temperature	2.72	3.046828e+11
MarkDown5	8.15	1.008018e+11
MarkDown1	8.26	9.442963e+10
MarkDown4	10.98	7.790066e+10
MarkDown3	6.25	4.945029e+10
HN	2.95	3.474087e+10
MarkDown2	3.65	2.034858e+10



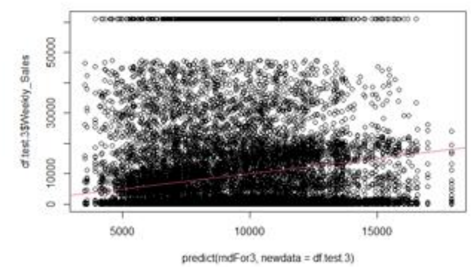


	%IncMSE	IncNodePurity
Size	38.68	936020447906
CPI	42.13	857245490351
Unemployment	25.59	439094332876
Week	3.20	385521980147
Temperature	10.69	148825222278
MarkDown5	5.51	49346371249
MarkDown1	6.76	48551087573
MarkDown4	7.60	30992652314
MarkDown3	8.20	30610899954
HN	0.00	15725005727
MarkDown2	5.62	11394093494





	%IncMSE	IncNodePurity
Week	-56.06	116742401111
Unemployment	9.35	71269813405
CPI	3.58	61889239835
Temperature	-17.16	43037513763
Size	5.58	29505217382
MarkDown5	-7.64	17035537643
MarkDown1	-7.00	14410775351
MarkDown3	-7.71	9101692873
MarkDown4	-4.69	5165085186
HN	-13.03	4781686334
MarkDown2	-3.08	2971804860



Random Forest Plots

COMPARISON TABLE

Going by the accuracy and output of all models, the Gradient Boosting Model and Decision tree models are most effective. These models clearly provide insights on how the external and internal variables are affecting the weekly sales. We learned that when the RMSE decreases, the model's performance improves.

1.

	GBM Model 1, Model 2, Model3	**Decision Trees** Model 1, Model 2	**Linear Regression** Model 1, Model 2
RMSE	18926.43 14212.84 15430.54	12852.76 13356.9	18100.23 14700.21 15567.09
Accuracy	81.37% 75.98% 89.65	88% 91%	71.59% 49.9% 46.3%

RECOMMENDATIONS

- Going by the accuracy and output of all models, the GBM, decision trees and random forest models are most accurate and provides insights on how the external and internal variables are affecting the weekly sales. Both RMSE and R2 indicate the goodness of the fit.
- Stores are making huge sales during holiday season, which is an important indicator for planning inventory and staff to handle this surge in demand during holiday season.
- By EDA, we can infer that type A store is the largest store and C is the smallest. Size of the Store is significantly affecting in overall sales, it is recommended to open new stores of Type A or enhance the area of existing ones
- External factors such as CPI,Fuel and Unemployment are also significantly affecting the Sales. Sales are higher at warmer temperatures. At higher CPI and Unemployment rate, Weekly Sales decreases
- Holiday and Store do not show significant relations; there is a residual boost in sales peak during the weeks surrounding the holidays. This can probably be attributed to promotions before and after the holiday itself however Department and Sales are significant as certain departments indicate higher sales compared to others.
- Fuel, Temperature, CPI are external data indicators important for estimating running cost to business.
- Markdowns are affecting mainly Type B and C stores. Markdown 3, 5 are most effective.