# CSC-433 PROJECT SUBMISSION

Data Mining

on

Big-Mart Sales Dataset

(Technical Report)

by

**Pradeep Sathyamurthy**

Under the guidance of: Prof. Steve D. Jost

DePaul University

08th June 2017

# Contents

# Project Summary:

## Introduction:

I have registered myself with AV (Analytic Vidhya) in India, which is like Kaggle in US, where Machine Learning hackathons are held. As part of its hackathon, AV had opened a challenge to predict revenue generated by a retail store called Big-Mart whose dataset contains properties of store and product being sold there. I would like to start with simple linear regression as part of this project and later advance with other machine learning algorithm to enhance my output in explaining the variance of sales with respective to other independent variables provided.

## Project Scope:

In this project my main interest is to apply the knowledge obtained from courses below to perform data mining on the Big-Mart dataset:

*CSC-433: Scripting for Data Analysis*

*CSC-423: Data Analysis and Regression*

*CSC-465: Data Visualization*

I will restrict my analysis by doing Exploratory Data Analysis (EDA), Treating missing values with decision tree and mean value imputation for factor and numerical variable respectively and finally apply simple OLS linear regression model on the cleaned dataset and observe the variance explained. I am planning to continue this project as part of my Summer courses where I will study about few regularization techniques like Ridge Regression and another advance data mining techniques like Random Forest, XGBOOST, etc., to make my model more sophisticated in explaining high variance with model being free from overfit or underfit.

## Dataset Description:

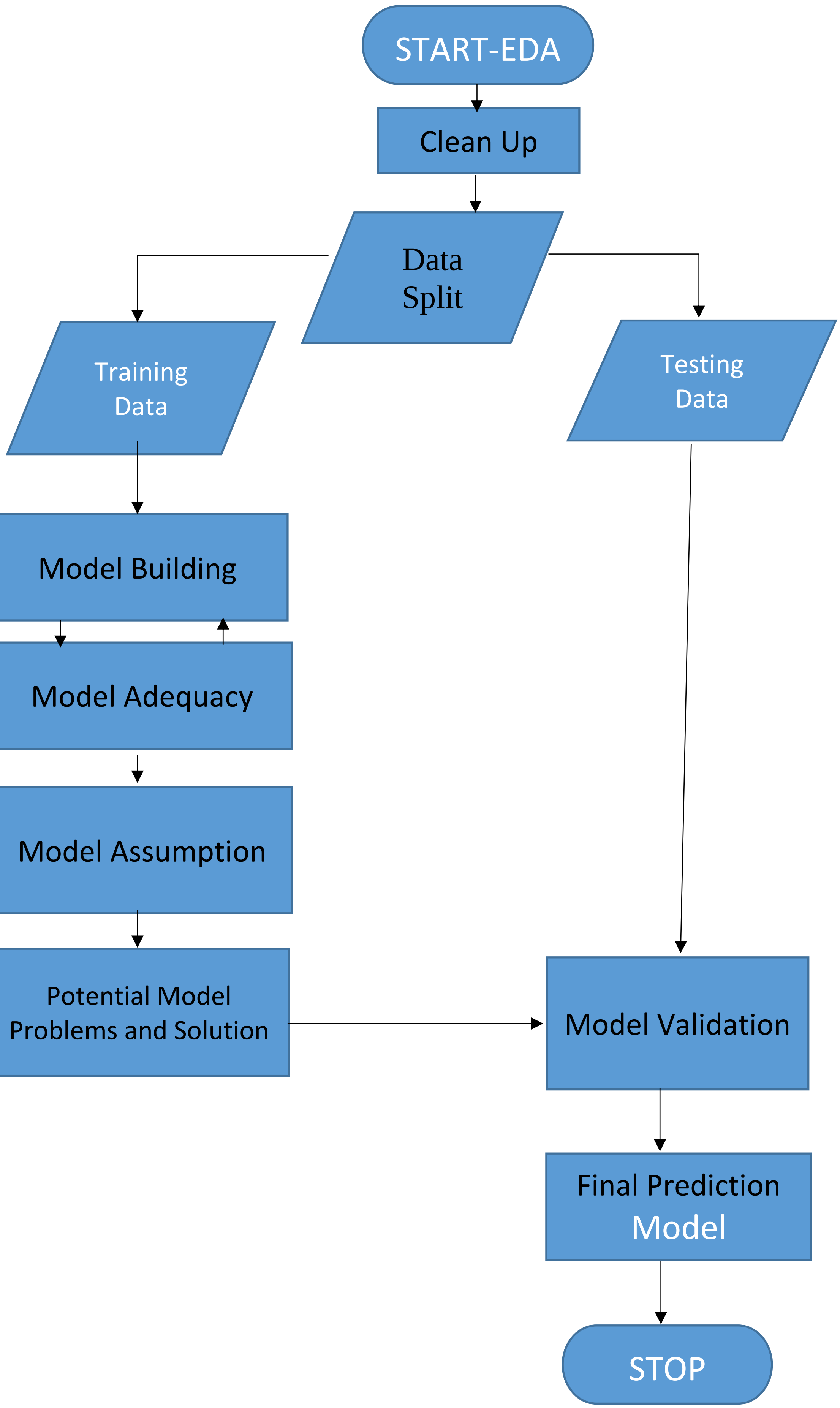
The data scientists at Big-Mart have collected 2013 sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim is to build a predictive model and find out the sales of each product at a store. Using this model, Big-Mart will try to understand the properties of products and stores which play a key role in increasing sales. This dataset is available on registration to participate in the hackathon conducted by AV through this link https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/

***Files:***

We were shared with two files train.csv and test.csv. Former to train the algorithm and later to validate the final algorithm built which do not have the values for dependent variable (count) for competition evaluation purpose by AV. However, for this project since we are requested to show and prove the final model behaviour, we have ***considered only train.csv file*** which has the information of dependent variable. Based on this train.csv file ***we created our Train [6113 records] and Test data [2410 records]*** for model building and model validation respectively. Thus, for this project purpose test.csv file shared by AV has been discarded. Below are features descriptions available as part of the dataset:

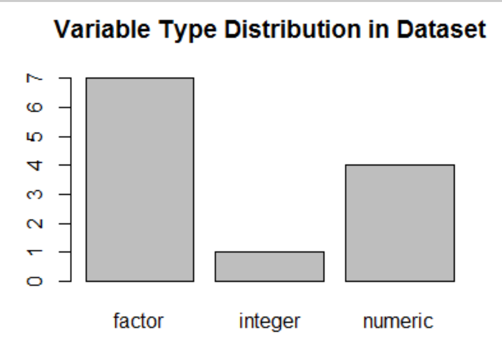
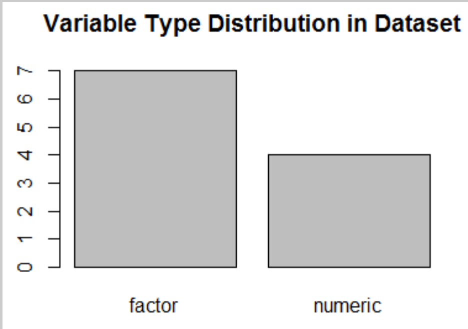
| SI.NO | Variable Name | Description |
|---|---|---|
| **1** | **Item_Identifier** | Unique product ID |
| **2** | **Item_Weight** | Weight of product |
| **3** | **Item_Fat_Content** | Whether the product is low fat or not |
| **4** | **Item_Visibility** | The % of total display area of all products in a store allocated to the product |
| **5** | **Item_Type** | The category to which the product belongs |
| **6** | **Item_MRP** | Maximum Retail Price (list price) of the product |
| **7** | **Outlet_Identifier** | Unique store ID |
| **8** | **Outlet_Establishment_Year** | The year in which store was established |
| **9** | **Outlet_Size** | The size of the store in terms of ground area covered |
| **10** | **Outlet_Location_Type** | The type of city in which the store is located |
| **11** | **Outlet_Type** | Whether the outlet is just a grocery store or some sort of supermarket |
| **12** | **Item_Outlet_Sales** *(Dependent Var)* | Sales of the product in the store. This is the outcome variable to be predicted. |

## Project Methodology Layout



# Exploratory Data Analysis

From an analysis point of view it is always wise to have data with minimum class differentiation. 'R' being built above an ancient language, data being in either Factor or Numeric form would serve the purpose of analysis better. So, we explored our dataset initially and tried to convert them into Factors and Numeric where ever required. Thus, we totally have 7 factor variables, 1 integer and 4 numeric variables as part of our Mart_Train.csv file.

| Before Datatype Conversion | After Datatype conversion and column pruning |
|---|---|
|  |  |





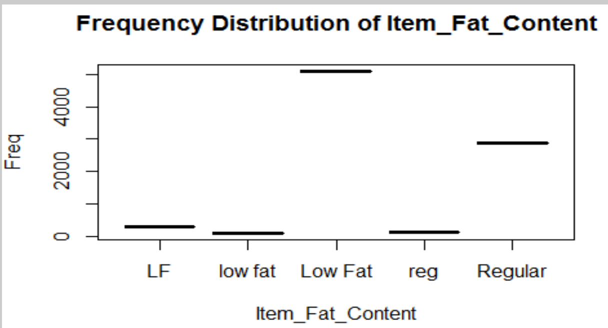
It is always wise to start a model building with hypothesis generation and an exploratory data analysis. This will help us to:

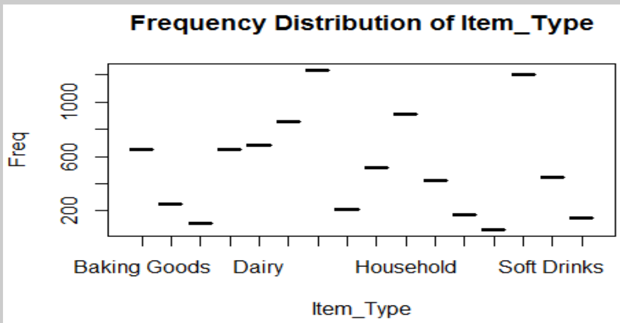
- Understand the relationship between the variables
- Gain domain expertise
- Avoid bias based samples
- Build a structure modelling with a structured approach.

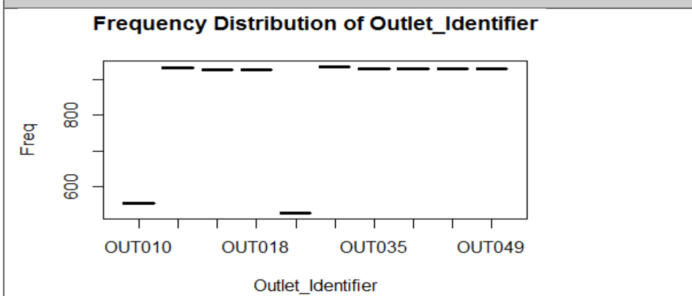
Best approach to validate these hypothesis is through visualization, below are few EDA[R] done on ***train.csv*** file:

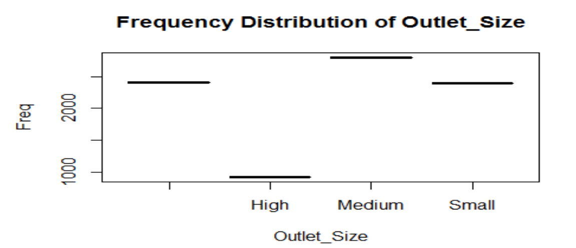
## EDA on Factor variables:

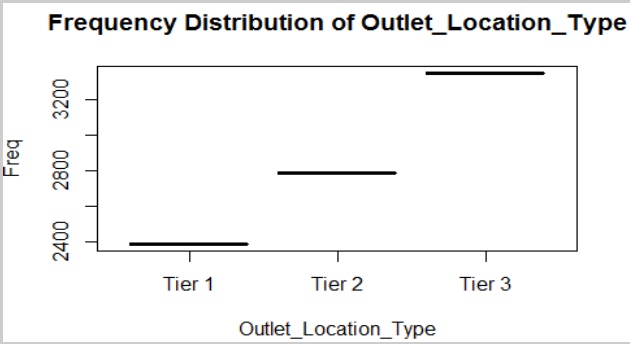
**Exploratory Data Analysis on Factor Variables Before Clean-up and Split**

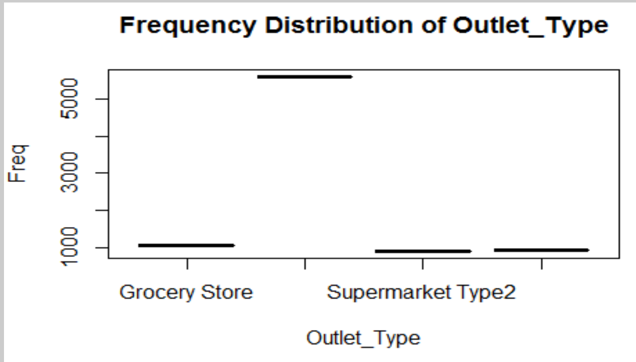


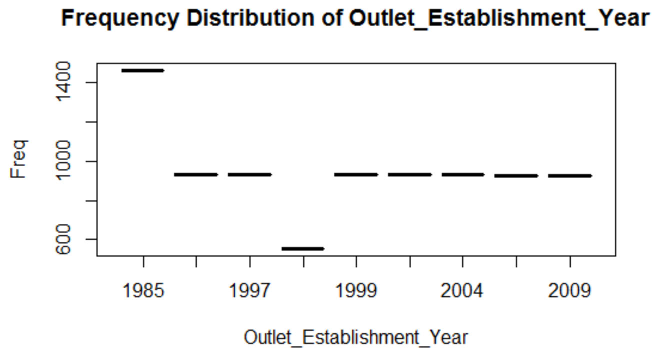





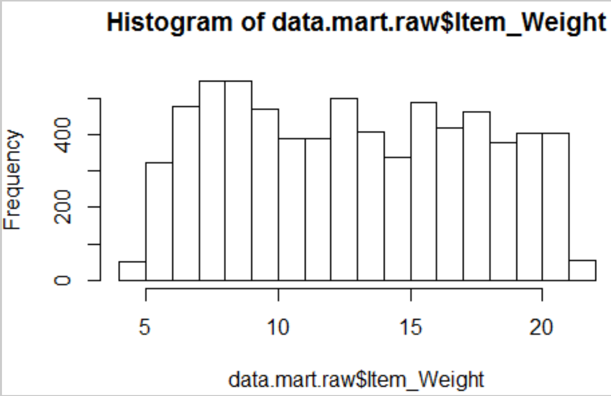



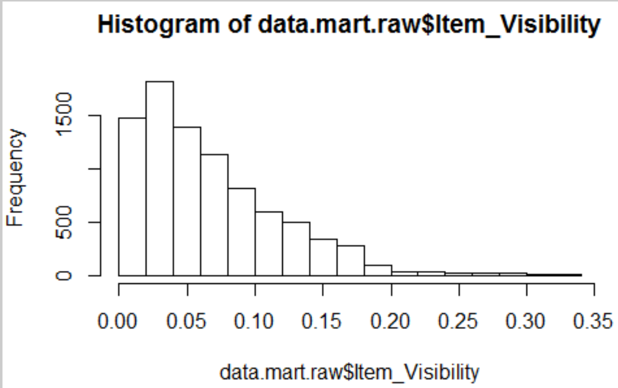

## Hypothesis:

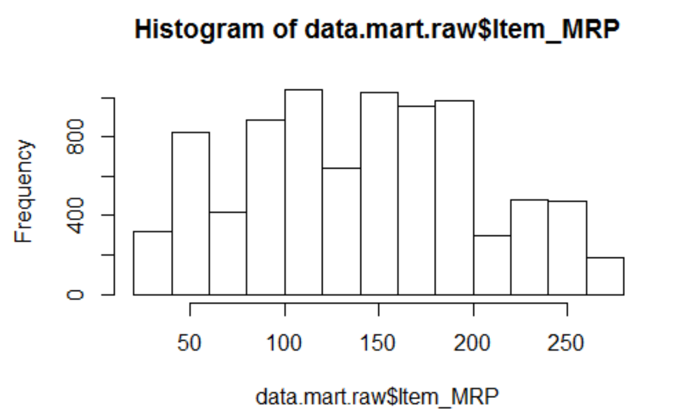
1. Low fat food is being purchased more compare to the regular fat foods
2. Food products like Fruits and Vegetables, snacks have higher sale; Households, canned, dairy and baking good have average sales and others are bought even less
3. OUT010 and OUT019 have lowest sale compare to others
4. Big mart owns Small and medium sized outlets more when compare to High size outlet
5. Big mart outlets are situated more in Tier3 and Tier2 locations when compare to Tier1 regions
6. Other than 1997, we could see a constant sale obtained in all years till 2009

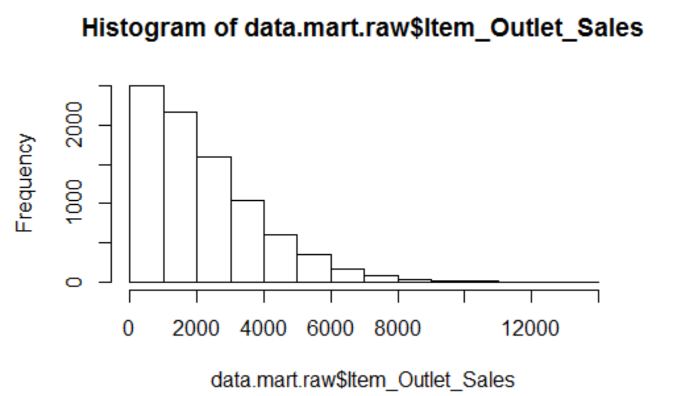
## EDA on Factor variables:

**Exploratory Data Analysis on Numerical Variables Before Clean-up and Split**




## Hypothesis:

1. Item weight has a normal distribution, which means product of all weight are available in store at equal proportion, it not just the whole sale which is happening in store

2. Product visibility is skewed to right, stores have more of small display area for product more and interestingly there is a size 0 which can be even online sold product

3. MRP of the product is also quite normally distributed, which means product of all price range from $31 to $266 is available in store in equal proportion, so it targets all kind of customers for its sales

4. Total sale revenue is skewed to right, meaning store constantly generate revenue of range $800 to $3000 in each of its outlet mostly

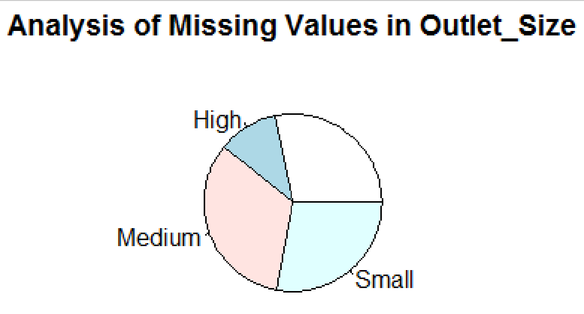
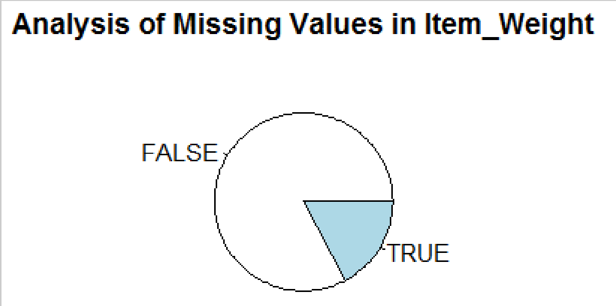
Thus, we hypothesized few scenarios based on our dataset, I would like to highlight same below:

*Groceries like fruit, vegetables and snacks with low fat content with minimum product visibility in a small and medium sized outlet situated in Tire-3 and Tier-2 region should generate revenue of at least $1000 to $3000.*

# Missing Value Treatment:

I tried to look at all variables with values being Na, NaN, NULL and blank. Through below visualization I could figure out there are 2 variables with missing values:

1. **Outlet_Size**
2. **Item_Weight**

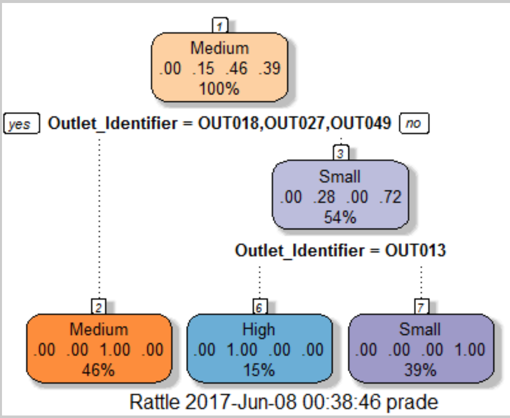
| Missing values in Factor Variables | Missing values in Numerical Variables |
|---|---|
|  |  |





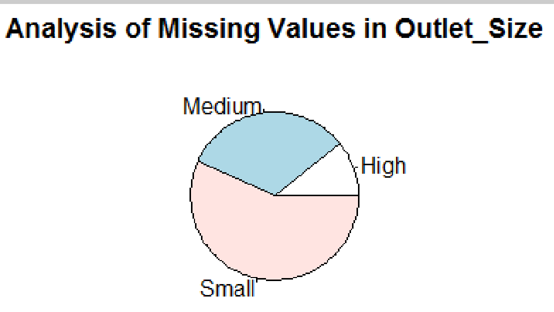
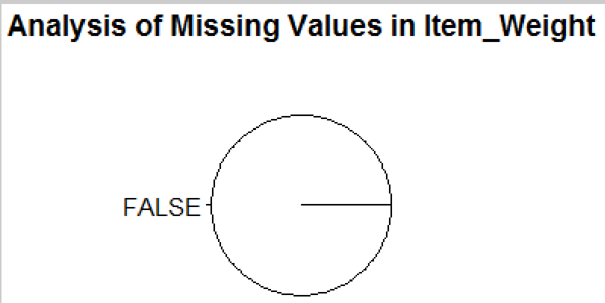
## Treating the Factor and Numerical Variable:

*Outlet_Size is a factor variable*, hence I decided to use decision tree to compute the missing values of Outlet_Size in the dataset. Thus, using below decision tree we imputed stores with outlet identifier OUT018, OUT027 and OUT049 as Medium type and if OUT013 it being of size High and rest all others as small.

| Decision Tree |
|---|
|  |



Similarly, I imputed the numerical variable Item_Weight using mean value of the them in an iterative state there by both median and mean was staying close to each other causing no bias in data. Post this I checked for missing values:

| Missing values in Factor Variables | Missing values in Numerical Variables |
|---|---|
|  |  |

## Data Split

In this dataset, we totally have 7 factor variables with 4 variables being a quantitative data. So, totally we have 11 variables (7 factor + 4 quantitative). As per thumb of rule we need to have at least 110 sample records to split the data in to train and test data. Since, we have totally 8523 records as part of our train.csv we can do a split[R] of ***Training-Data: Testing-Data = 80:20 ratios*** through which ***Training data can be used to build our model while Testing Data can be used to test our model for model validation and prediction***.

### Training Data:

From here on we will refer our Training dataset with name ***data.train*** which is 80% of Simple Random Sampled data from train.csv file. Data.train has a total number of 8710 samples through which we will be building and training our model.

*Training Dataset: data.train = 80% of (8523 samples of Mart_train.csv file)*

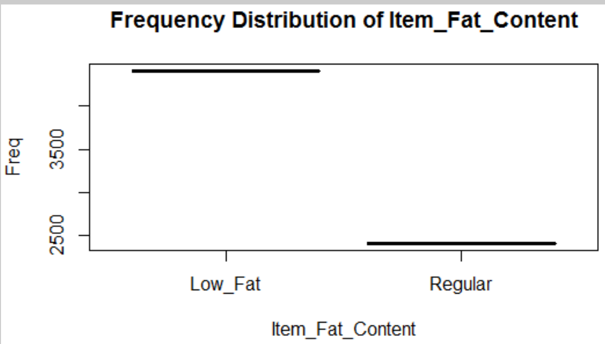
### Test Data:

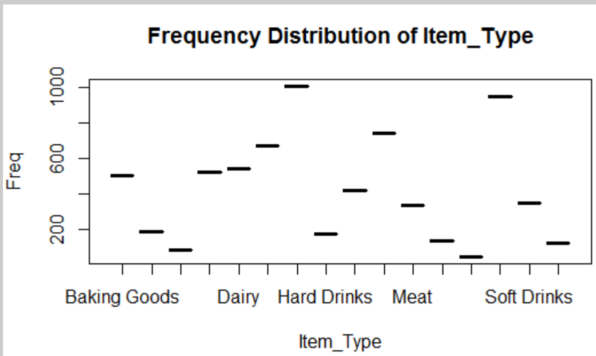
From here on we will refer our Testing dataset with name ***data.test*** which is the remaining 20% of Simple Random Sampled data from train.csv file. Data.test has a total number of 1703 samples through which we will be validating our model and subject it for prediction. Since, our testing dataset is a remaining sample left out by training data, ***our data.test is no way a subset of data.train these are two simple random samples of train.csv***.

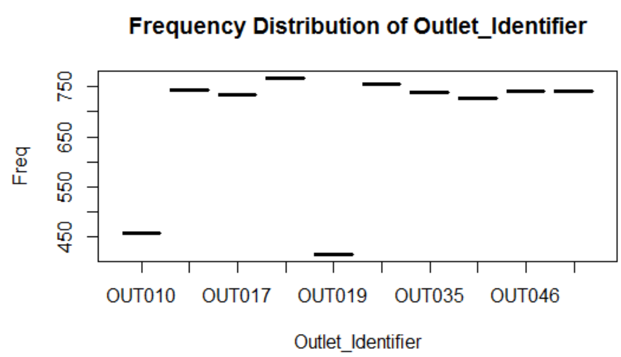
*Testing Dataset: data.test = 20% of (remaining 8523 samples of Mart_train.csv file)*

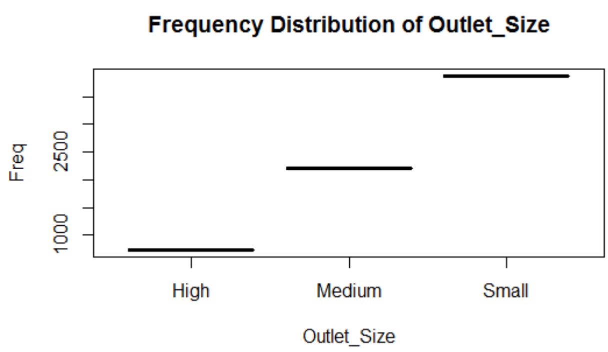
### Exploratory Data Analysis on data.train after clean-up:

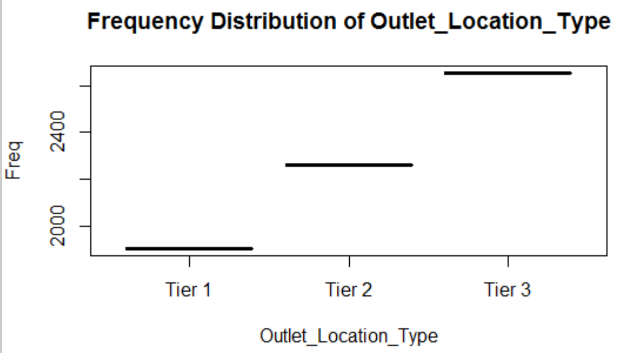
**Exploratory Data Analysis on Factor Variables After Clean-up and Split**

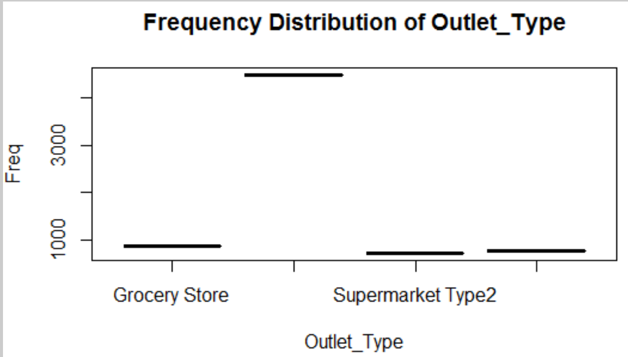


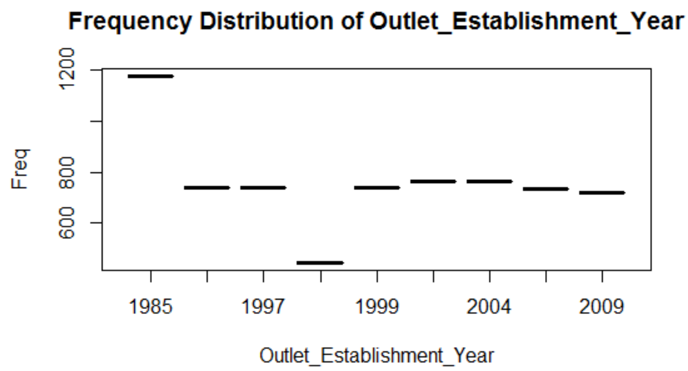



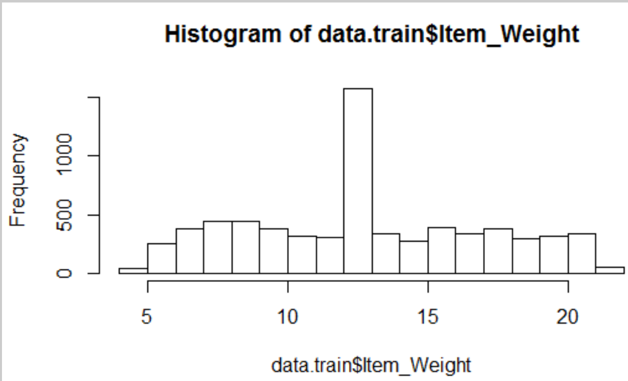



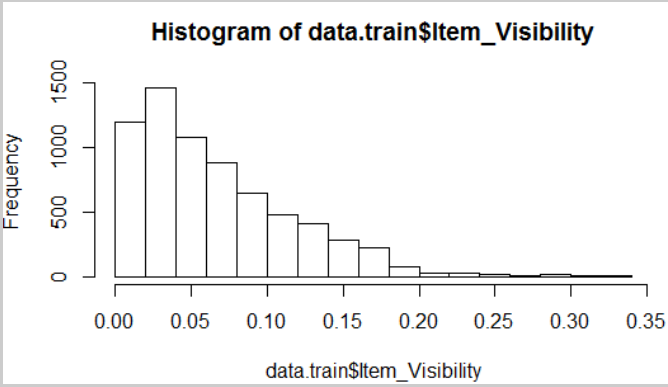


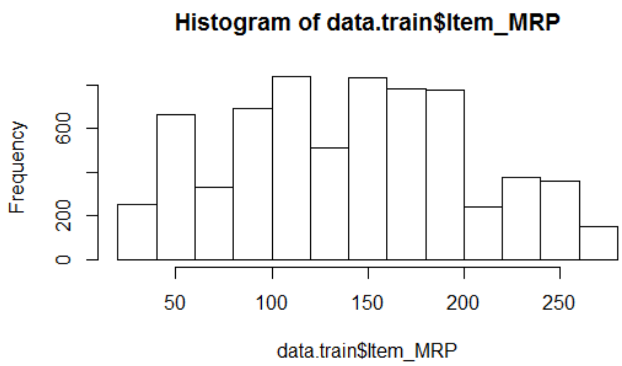



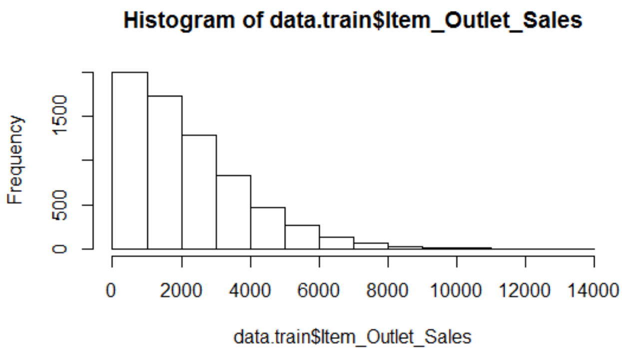




## Exploratory Data Analysis on Numerical Variables After Clean-up and Split




From above graphs we could infer that data clean up on factor variable has been done to derive variables with minimum class labels and clean up on Numerical variables either turned the data normally distributed to certain extent. Now we can take this dataset for Model building.

# Model Building and Adequacy:

```
Call:
lm(formula = Item_Outlet_Sales ~ Item_Fat_Content + Item_Type + 
    Outlet_Identifier + Outlet_Establishment_Year + Outlet_Size + 
    Outlet_Location_Type + Outlet_Type + Item_Weight + Item_Visibility + 
    Item_MRP, data = data.train)

Residuals:
    Min      1Q  Median      3Q     Max 
-3876.7  -680.3   -88.8   572.4  7936.2 

Coefficients: (15 not defined because of singularities)
                                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)                    -1848.1240    92.5926 -19.960   <2e-16 ***
Item_Fat_ContentRegular           16.0242    31.4328   0.510    0.610    
Item_TypeBreads                   38.6307    93.6021   0.413    0.680    
Item_TypeBreakfast                59.7512   129.4172   0.462    0.644    
Item_TypeCanned                   99.0539    69.5947   1.423    0.155    
Item_TypeDairy                    -7.2979    69.2438  -0.105    0.916    
Item_TypeFrozen Foods              7.1309    65.1340   0.109    0.913    
Item_TypeFruits and Vegetables    75.9955    61.0891   1.244    0.214    
Item_TypeHard Drinks              86.2710   104.4468   0.826    0.409    
Item_TypeHealth and Hygiene       21.4118    76.6668   0.279    0.780    
Item_TypeHousehold               -48.1652    66.2863  -0.727    0.467    
Item_TypeMeat                     72.4464    79.1534   0.915    0.360    
Item_TypeOthers                  -10.7573   109.9311  -0.098    0.922    
Item_TypeSeafood                 164.5292   166.9453   0.986    0.324    
Item_TypeSnack Foods              27.2848    61.1920   0.446    0.656    
Item_TypeSoft Drinks             -36.2746    78.2460  -0.464    0.643    
Item_TypeStarchy Foods           112.2731   114.2634   0.983    0.326    
Outlet_IdentifierOUT013         1915.4365    68.7148  27.875   <2e-16 ***
Outlet_IdentifierOUT017         1978.0706    68.7015  28.792   <2e-16 ***
Outlet_IdentifierOUT018         1632.1324    69.0396  23.641   <2e-16 ***
Outlet_IdentifierOUT019            4.0058    76.5855   0.052    0.958    
Outlet_IdentifierOUT027         3378.9020    68.5778  49.271   <2e-16 ***
Outlet_IdentifierOUT035         2020.7990    68.2467  29.610   <2e-16 ***
Outlet_IdentifierOUT045         1818.4370    68.3390  26.609   <2e-16 ***
Outlet_IdentifierOUT046          1896.4968    68.6683  27.618   <2e-16 ***
Outlet_IdentifierOUT049          1974.2156    68.6753  28.747   <2e-16 ***
Outlet_Establishment_Year1987           NA         NA      NA       NA    
Outlet_Establishment_Year1997           NA         NA      NA       NA    
Outlet_Establishment_Year1998           NA         NA      NA       NA    
Outlet_Establishment_Year1999           NA         NA      NA       NA    
Outlet_Establishment_Year2002           NA         NA      NA       NA    
Outlet_Establishment_Year2004           NA         NA      NA       NA    
Outlet_Establishment_Year2007           NA         NA      NA       NA    
Outlet_Establishment_Year2009           NA         NA      NA       NA    
Outlet_SizeMedium                       NA         NA      NA       NA    
Outlet_SizeSmall                        NA         NA      NA       NA    
Outlet_Location_TypeTier 2              NA         NA      NA       NA    
Outlet_Location_TypeTier 3              NA         NA      NA       NA    
Outlet_TypeSupermarket Type1            NA         NA      NA       NA    
Outlet_TypeSupermarket Type2            NA         NA      NA       NA    
Outlet_TypeSupermarket Type3            NA         NA      NA       NA    
Item_Weight                        -0.1743     3.2540  -0.054    0.957    
Item_Visibility                  -201.0298   276.1965  -0.728    0.467    
Item_MRP                           15.6280     0.2220  70.395   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1127 on 6791 degrees of freedom
Multiple R-squared:  0.5675,	Adjusted R-squared:  0.5657 
F-statistic: 318.2 on 28 and 6791 DF,  p-value: < 2.2e-16
```

- Over all model was significant with p-value < 0.01
- No significant correlation exists with all numerical variables available
- Variables Item_Fat_Content, Outlet_Identifier and Item_MRP seem significant variables with respective p-value < 0.05
- However, ***the variance that this model could explain was only 56.57% which is not efficient***
- We applied even the stepwise algorithm which even explains only 56.66% of variance in sales.

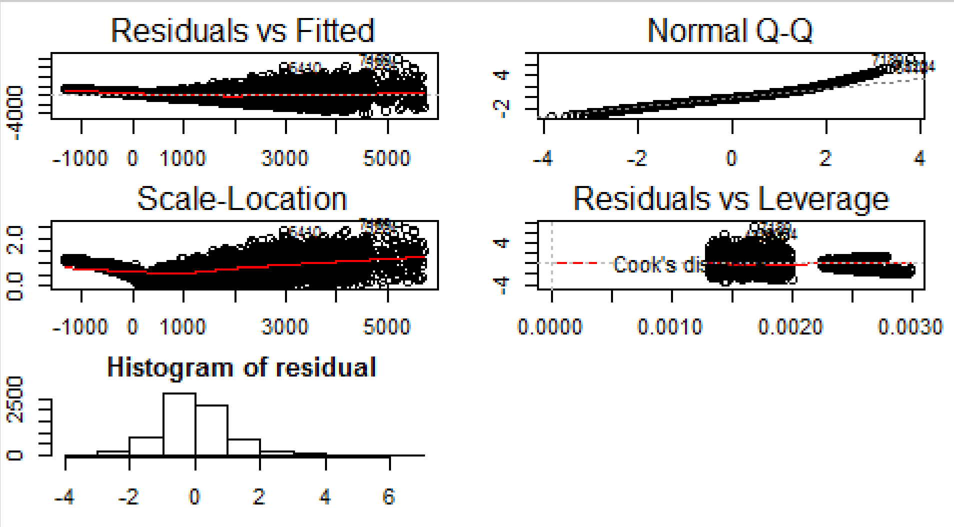
## Model Assumption

We start to create any models with few assumptions, in which two major assumptions are:

1. **All pairs of error terms are not correlated that is error terms are independent to each other**
2. **Error is normally distributed with mean=0 and Standard Deviation being constant**

So, to re-confirm that our assumptions hold good for the model that would be subjected to prediction, we need to perform few residual analyses before concluding the final model.

**Model: Item_Outlet_Sales ~ Outlet_Identifier + Item_MRP**




### Residual Analysis

1. Graph plotted between Residual and Fitted is used to confirm assumption-1
2. We expect residual plot with no trends or pattern. From above figure we could infer that ***there is a concrete trend that exists*** in this plot.
3. ***There is a dramatic increase in variability***

### Heteroscedasticity:

1. We say a model as heteroscedastic when there is no constant variance. Funnel shape of residual plot clearly identifies the model is heteroscedastic.
2. Even in our residual graph we see a funnel shape and can say that ***our model is heteroscedastic*** in nature.

### Normal Probability Plot:

1. From normality plot for the residual, we can notice that most of the points fall reasonably close to straight line which indicates that ***normality assumption is satisfied***.

### Outlier and influential Points:

1. Residual vs Leverage graph infers that there are few influential or outliers present.
2. Hence, we calculated for observations which are considered as outliers based on model built, any studentized residual greater than 3 or less than -3 where considered as outliers. We obtained a list of 74 observations.
3. We also wanted to find observations which are influential based on H-hat method. We got a cut off 0.01319648 and hence considered any value above this as influential point. However, our model fetched only one observation.
4. We compared list of outliers with influential point and there was just 1 matching record in observation 831 which was removed. While the looking at other observations we can concluded that rest of the 73 observations obtained are ***natural outliers and removing them will either over-fit or under fit the model***.

## Potential Model Problem and Solution:

Above residual analysis clearly indicates that our model is suffering from heteroscedasticity, that is a state with non-constant variance. Thus, we cannot use this model directly for prediction or model validation. We need to fix this. One possible solution is to try transforming the dependent variable and see how our model behaves with respective to explaining variance and residual behaviour. So, we tried to perform model transformation on model selected from model adequacy for residual analysis.

### Model Transformation:

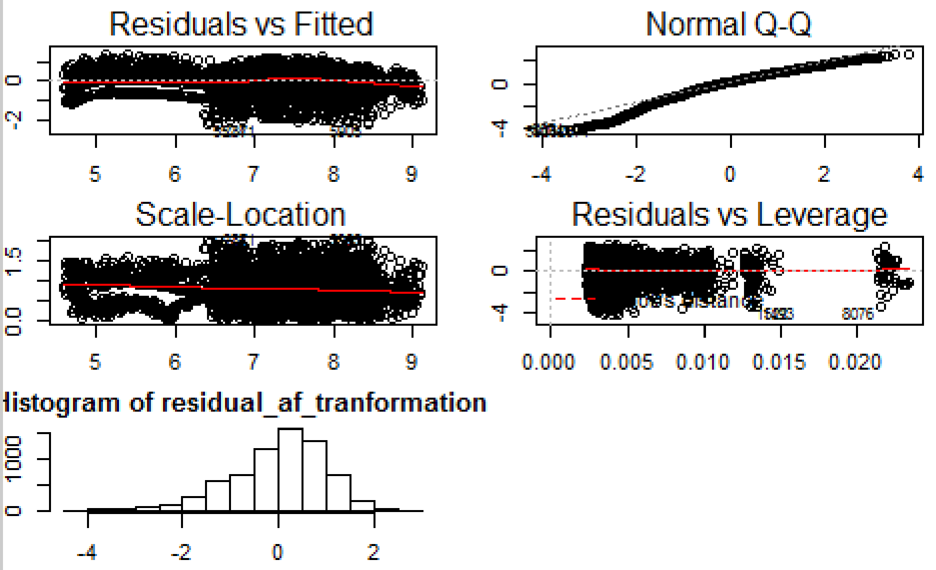
Since the normal probability plot show a s-shape, we assumed log transformation will do a great difference. Further based on EDA done, we found that **doing log transformation on sale will have a significant impact** on outliers/influential points. Also, the heteroscedastic nature force us to apply a transformation to make it homoscedastic.

```
> summary(model3_transformed)

Call:
lm(formula = log(Item_Outlet_Sales) ~ Item_Fat_Content + Item_Type +
    Outlet_Identifier + Outlet_Establishment_Year + Outlet_Size +
    Outlet_Location_Type + Outlet_Type + Item_Weight + Item_Visibility +
    Item_MRP, data = data.train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.24677 -0.28888  0.07259  0.37362  1.34912

Coefficients: (15 not defined because of singularities)
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                   4.3542772  0.0441451  98.635   <2e-16 ***
Item_Fat_ContentRegular       0.0004711  0.0149862   0.031   0.9749
Item_TypeBreads               0.0459401  0.0446264   1.029   0.3033
Item_TypeBreakfast           -0.0240762  0.0617019  -0.390   0.6964
Item_TypeCanned               0.0473303  0.0331805   1.426   0.1538
Item_TypeDairy               -0.0575587  0.0330132  -1.744   0.0813 .
Item_TypeFrozen Foods        -0.0312408  0.0310538  -1.006   0.3144
Item_TypeFruits and Vegetables  0.0128284  0.0291253   0.440   0.6596
Item_TypeHard Drinks          0.0112168  0.0497968   0.225   0.8218
Item_TypeHealth and Hygiene   0.0302966  0.0365522   0.829   0.4072
Item_TypeHousehold           -0.0381760  0.0316032  -1.208   0.2271
Item_TypeMeat                 0.0458601  0.0377378   1.215   0.2243
Item_TypeOthers               0.0377424  0.0524116   0.720   0.4715
Item_TypeSeafood             -0.0230763  0.0795941  -0.290   0.7719
Item_TypeSnack Foods          0.0203009  0.0291744   0.696   0.4865
Item_TypeSoft Drinks         -0.0181766  0.0373052  -0.487   0.6261
Item_TypeStarchy Foods        0.0076445  0.0544771   0.140   0.8884
Outlet_IdentifierOUT013       1.9448733  0.0327610  59.366   <2e-16 ***
Outlet_IdentifierOUT017       1.9991461  0.0327547  61.034   <2e-16 ***
Outlet_IdentifierOUT018       1.7985625  0.0329158  54.641   <2e-16 ***
Outlet_IdentifierOUT019       0.0266101  0.0365135   0.729   0.4662
Outlet_IdentifierOUT027       2.5034020  0.0326957  76.567   <2e-16 ***
Outlet_IdentifierOUT035       2.0130805  0.0325378  61.869   <2e-16 ***
Outlet_IdentifierOUT045       1.9243616  0.0325818  59.062   <2e-16 ***
Outlet_IdentifierOUT046       1.9661162  0.0327388  60.055   <2e-16 ***
Outlet_IdentifierOUT049       2.0098021  0.0327422  61.383   <2e-16 ***
Outlet_Establishment_Year1987        NA         NA      NA       NA
Outlet_Establishment_Year1997        NA         NA      NA       NA
Outlet_Establishment_Year1998        NA         NA      NA       NA
Outlet_Establishment_Year1999        NA         NA      NA       NA
Outlet_Establishment_Year2002        NA         NA      NA       NA
Outlet_Establishment_Year2004        NA         NA      NA       NA
Outlet_Establishment_Year2007        NA         NA      NA       NA
Outlet_Establishment_Year2009        NA         NA      NA       NA
Outlet_SizeMedium                    NA         NA      NA       NA
Outlet_SizeSmall                     NA         NA      NA       NA
Outlet_Location_TypeTier 2           NA         NA      NA       NA
Outlet_Location_TypeTier 3           NA         NA      NA       NA
Outlet_TypeSupermarket Type1         NA         NA      NA       NA
Outlet_TypeSupermarket Type2         NA         NA      NA       NA
Outlet_TypeSupermarket Type3         NA         NA      NA       NA
Item_Weight                  -0.0008590  0.0015514  -0.554   0.5798
Item_Visibility               0.0252753  0.1316815   0.192   0.8478
Item_MRP                      0.0084052  0.0001058  79.410   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5374 on 6791 degrees of freedom
Multiple R-squared:  0.7253,    Adjusted R-squared:  0.7241
F-statistic: 640.2 on 28 and 6791 DF,  p-value: < 2.2e-16
```

- Over all model was significant with p-value < 0.01
- Predictors Outlet_Identifier and ITEM_MRP is the only variable being significant with p-value<0.05
- Though there is a ***improvement in Adjusted R-Square value from 0.5667 to 0.7241. However, since there is less variable explaining the Sales, we might get Rank issue and data might underfit when subjected to other dataset.***
- Though we assure efficiency into this model using transformation, it should be deceiving which can be concluded with model validation. However, after transformation this model ***explains about 71.41% of variance*** in sales data of Big-Mart store.

## Residual Analysis on Transformed Model

**Model: log(Item_Outlet_Sales) ~ Outlet_Identifier + Item_MRP**




- From above residual graph we can infer that there I ***no more pattern exist*** which indicates no correlation with Residual and Fitted. This ***satisfies our assumption-1*** of error terms to be independent to each other

- From the same residual plot, we also see the ***funnel shape no more exists*** and hence can be proved that ***model is no more suffering from Heteroscedasticity***. Model is now Homoscedastic.

- S-shape in normal probability plot is also corrected to some extent which means our error terms are normally distributed. This ***satisfies our assumption-2*** of error being normally distributed.

- Check on outlier and Influential point was also done which again proved that they ***are natural outliers*** in the system and can be treated as it is in the data.

# Model Validation:

This is the final stage in building an analytical model. This validation will confirm the following:

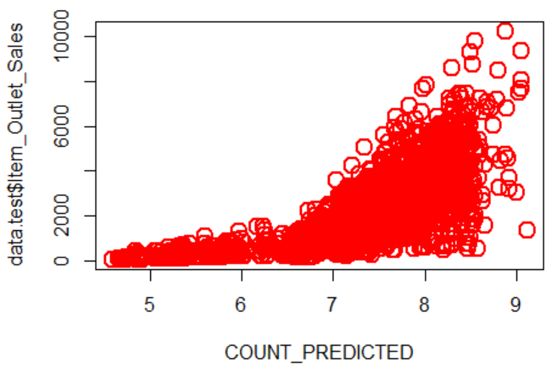
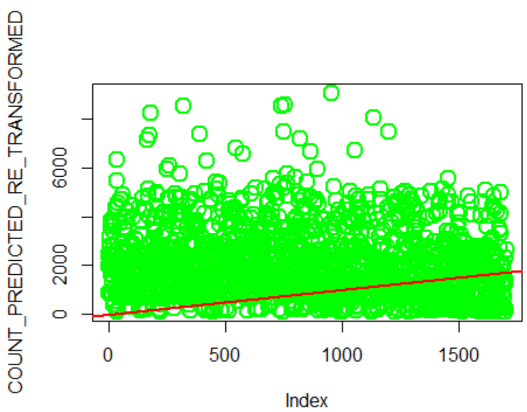
1. Is the model over fitting?
2. Is the model under fitting?
3. Is the model good for all samples of the population?

In order to test our model log transformed model we will be using the data.test which we had split as part of initial data split.

```
> str(data.test)
'data.frame':	1703 obs. of  11 variables:
 $ Item_Weight              : num  17.5 12.9 9 12.9 12.9 ...
 $ Item_Fat_Content         : Factor w/ 2 levels "Low_Fat","Regular": 1 1 2 1 2 2 1 1 2 1 ...
 $ Item_Visibility          : num  0.0168 0.1275 0.0692 0.0342 0.0354 ...
 $ Item_Type                : Factor w/ 16 levels "Baking Goods",..: 11 14 3 8 1 6 15 9 7 14 ...
 $ Item_MRP                 : num  141.6 107.8 54.4 113.3 144.5 ...
 $ Outlet_Identifier        : Factor w/ 10 levels "OUT010","OUT013",..: 10 6 10 6 6 9 9 10 6 6 ..
 $ Outlet_Establishment_Year: Factor w/ 9 levels "1985","1987",..: 5 1 5 1 1 3 3 5 1 1 ...
 $ Outlet_Size              : Factor w/ 3 levels "High","Medium",..: 2 2 2 2 2 3 3 2 2 2 ...
 $ Outlet_Location_Type     : Factor w/ 3 levels "Tier 1","Tier 2",..: 1 3 1 3 3 1 1 1 3 3 ...
 $ Outlet_Type              : Factor w/ 4 levels "Grocery Store",..: 2 4 2 4 4 2 2 2 4 4 ...
 $ Item_Outlet_Sales        : num  2097 4023 718 2304 4064 ...
```

## Model Performance through Prediction

As part of this process we will inject our data.test into our log transformed model and look at its prediction. Since, we have done log transformation on dependent variable “count”, ***output from prediction needs to be subjected to exponent to get the final prediction value for count variable***. Below are the graphs of prediction performance of our model without exponentiation and with exponentiation.

| Prediction Performance on Log Transformed Model | Prediction Performance on model after exponentiation |
| --- | --- |
|  |  |





- Above graph in red shows the model performance for the log transformed model
- While graph in blue shows the model performance for final model which was re-tansformed
- Final model performance proves that ***though the final model explains 72% of variance it behaves badly when it comes for the prediction of Sales data.***

## Model Prediction with Confidence and Prediction Interval values:

Below is the snapshot of model performance, it consists of below details:

1. Actual Count Value
2. Predicted Count Value
3. Predicted Interval Low
4. Predicted Interval High
5. Confidence Interval Low
6. Confidence Interval High

rstudio

| | data.test.Item_Outlet_Sales | data.test.predicted_count | data.test.prediction_interval_low | data.test.prediction_interval_high | data.test.confidence_interval_low | data.test.confidence_interval_high |
|---|---|---|---|---|---|---|
| 1 | 2097.2700 | 1893.4055 | 659.47508 | 5436.1182 | 1821.4005 | 1968.2571 |
| 2 | 4022.7636 | 2341.7013 | 815.60998 | 6723.2686 | 2252.1252 | 2434.8402 |
| 3 | 718.3982 | 910.7689 | 317.17235 | 2615.2971 | 872.6127 | 950.5935 |
| 4 | 2303.6680 | 2452.6901 | 854.27275 | 7041.8829 | 2359.2869 | 2549.7910 |
| 5 | 4064.0432 | 3187.9540 | 1110.38087 | 9152.7612 | 3067.7159 | 3312.9049 |
| 6 | 4078.0250 | 2202.9344 | 767.27704 | 6324.8600 | 2118.6152 | 2290.6093 |
| 7 | 2085.2856 | 2392.7138 | 833.36730 | 6869.8152 | 2300.4255 | 2488.7046 |
| 8 | 3791.0652 | 1976.0238 | 688.25085 | 5673.3240 | 1900.8571 | 2054.1628 |
| 9 | 2797.6916 | 2776.4855 | 967.06191 | 7971.4357 | 2671.5884 | 2885.5013 |
| 10 | 2180.4950 | 1977.7074 | 688.80801 | 5678.3990 | 1900.3128 | 2058.2542 |
| 11 | 3435.5280 | 3819.4758 | 1330.11592 | 10967.7623 | 3659.1679 | 3986.8067 |
| 12 | 2150.5340 | 1553.9450 | 541.23812 | 4461.5205 | 1494.6619 | 1615.5794 |
| 13 | 6258.5200 | 3777.3452 | 1315.31893 | 10847.8151 | 3610.5611 | 3951.8335 |
| 14 | 796.9626 | 794.3200 | 276.60193 | 2281.0549 | 759.8804 | 830.3204 |
| 15 | 3185.1872 | 3710.5747 | 1292.08043 | 10655.9657 | 3547.4983 | 3881.1477 |
| 16 | 484.7024 | 1307.5474 | 455.33093 | 3754.8080 | 1251.5437 | 1366.0572 |
| 17 | 3435.5280 | 2300.2779 | 801.17718 | 6604.3796 | 2211.9064 | 2392.1800 |
| 18 | 599.2200 | 887.5380 | 309.09001 | 2548.5253 | 850.8834 | 925.7716 |
| 19 | 2290.3520 | 1436.2160 | 500.22899 | 4123.5441 | 1381.1118 | 1493.5187 |
| 20 | 1427.4752 | 2504.8470 | 872.41561 | 7191.8226 | 2407.7188 | 2605.8933 |
| 21 | 583.2408 | 406.6275 | 141.54499 | 1168.1512 | 385.8461 | 428.5283 |
| 22 | 3285.7230 | 3508.3647 | 1221.97579 | 10072.7224 | 3375.6048 | 3646.3460 |
| 23 | 3185.8530 | 2770.7205 | 964.92792 | 7955.9228 | 2656.9284 | 2889.3862 |
| 24 | 2247.7408 | 1764.9643 | 614.73823 | 5067.3582 | 1697.7970 | 1834.7888 |
| 25 | 679.1160 | 526.3025 | 183.18710 | 1512.0840 | 498.5356 | 555.6159 |
| 26 | 699.0900 | 3880.6604 | 1351.41389 | 11143.5330 | 3717.1556 | 4051.3572 |
| 27 | 176.4370 | 350.5977 | 122.04537 | 1007.1563 | 332.9046 | 369.2312 |

Showing 1 to 28 of 1,703 entries

- ✓ Box in green color highlights the ***closely matched values***
- ✓ Box in red color highlights the count values falling into the closest prediction interval range
- ✓ However, this model is seriously underfitting the data, which cannot be applied for prediction.

## Final Proposed Model out of regression but this is subjected to further research and enhancement:

| ***Final Model to obtain a regression equation:*** |
|---|
| log(Item_Outlet_Sales) ~ Outlet_Identifier + Item_MRP |
| ✓ This model is seriously underfitting the data, which cannot be applied and advised for prediction.<br>✓ ***I would like to subject this dataset with other algorithms like factor analysis, SVM, Random Forest and XGBOOST to observe if I can get an optimized result out of it.*** |

## Timetable

| Phases | Description of Work | Start and End Dates |
|---|---|---|
| **Phase One** | Obtaining Dataset from Kaggle | 27-May-2017 to 28-May-2017 |
| **Phase Two** | Performing EDA on Big-Mart Dataset | 27-May-2017 to 31-May-2017 |
| **Phase Three** | Basic Model Building | 01-Jun-2017 to 02-Jun-2017 |
| **Phase Four** | Testing and Validation | 03-Jun-2017 to 04-Jun-2017 |
| **Phase Five** | Report Writing and Review | 05-Jun-2017 to 08-Jun-2017 |
| **Phase Six** | Deliverable Submission | 08-Jun-2017 |
| | | |
| | | |
| | | |

## Key Personnel

| | |
|---|---|
| Team Member | Pradeep Sathyamurthy |
| Professor | Prof. Steve D. Jost |
| Project for | CSC-433 |
| Target Team | DePaul CDM |

## Deliverables

| | | |
|---|---|---|
| Final Report | **Prady_CSC_423_Technical_Report.pdf** | Contains final Technical Report |
| Raw Data Set | **train.csv** | Raw Dataset downloaded from Kaggle |
| R | **Prady_Source_Files_Bike_Share.R** | Source File to Run through |
| R_Data_Files | **Prady_Project_All_Outcomes.RData** | Can be loaded in R to test all o/p |
| | | |

## R-Code for Big-Mart Dataset

```
##########################################################################################
# Author: Pradeep Sathyamurthy
# Date: 07-June-2017
# Course: CSC-433
# Guiding Prof: Prof. Steve Jost
# Project: Final Project Submission
# Train Dataset Name: mart_train.csv
# Test Dataset Name: mart_test.csv
##########################################################################################

# Libraries imported for this analysis
require(ggplot2) # <- needed for graphing
require(rpart) # <- Needed for building decision tree
require(rattle) # <- Needed to make decision tree look neat
require(rpart.plot) # <- Needed to make decision tree look neat
require(RColorBrewer) # <- Needed to make decision tree look neat
require(caret) # <- Needed for data splitting
require(MASS) # <- Needed for Outlier and Influential points detection
require(car) # Needed for Multicolinearity
```

```
# Step-1: Reading the trianing dataset
setwd("C:/Users/prade/Documents/GitHub/university_projects/BigMart_Sales_Prediction_With_Dimentionality_
Reduction")
data.mart.raw <- read.csv("Dataset/Mart_Train.csv")
head(data.mart.raw)

# Step-2: Researching the variables present
col_mart_name <- colnames(data.mart.raw) # <- Column names
col_mart_length <- length(col_mart_name) # <- There are 12 variables
var_det <- data.frame(Var_Name="NULL",Var_Type="NULL",stringsAsFactors = FALSE)
for(i in 1:col_mart_length){
  var_det <- rbind(var_det, c(colnames(data.mart.raw[i]),class(data.mart.raw[[i]])))
}
var_det <- var_det[-c(1),]
plot_var_type <- data.frame(table(var_det$Var_Type))
barplot(plot_var_type$Freq,names.arg = plot_var_type$Var1, main = "Variable Type Distribution in Dataset")
print(var_det,row.names = FALSE)
# above for loop says there are:
# 7 Factor Variables: Item_Identifier, Item_Fat_Content, Item_Type, Outlet_Identifier, Outlet_Size,
Outlet_Location_Type, Outlet_Type
# 1 integer variable: Outlet_Establishment_Year
# 4 Numeric variables: Item_Weight, Item_Visibility, Item_MRP, Item_Outlet_Sales


# Step-3: Converting the object type based on their values
# From the data we could conclude to have Item_Identifier as a ID variable and Outlet_Establishment_Year as a
factor
#data.mart.raw$Item_Identifier <- as.character(data.mart.raw$Item_Identifier)
data.mart.raw <- data.mart.raw[-c(1)]
head(data.mart.raw)
data.mart.raw$Outlet_Establishment_Year <- as.factor(data.mart.raw$Outlet_Establishment_Year)
summary(data.mart.raw)
col_mart_name <- colnames(data.mart.raw) # <- Column names
col_mart_length <- length(col_mart_name) # <- There are 12 variables
var_det <- data.frame(Var_Name="NULL",Var_Type="NULL",stringsAsFactors = FALSE)
for(i in 1:col_mart_length){
  var_det <- rbind(var_det, c(colnames(data.mart.raw[i]),class(data.mart.raw[[i]])))
}
var_det <- var_det[-c(1),]
plot_var_type <- data.frame(table(var_det$Var_Type))
barplot(plot_var_type$Freq,names.arg = plot_var_type$Var1, main = "Variable Type Distribution in Dataset")
print(var_det,row.names = FALSE)


# Step-4: Exploratory Data Analysis on factor variables
# After conversion below are factor variables:
# 1. Item_Fat_Content
# 2. Item_Type
# 3. Outlet_Identifier
# 4. Outlet_Size
# 5. Outlet_Location_Type
# 6. Outlet_Type
# 7. Outlet_Establishment_Year
# Let us plot these data to see the frequency of occurence
data.frame(table(data.mart.raw$Item_Fat_Content))
plot(data.frame(table(data.mart.raw$Item_Fat_Content)), main="Frequency Distribution of
Item_Fat_Content",xlab="Item_Fat_Content")
```

```
data.frame(table(data.mart.raw$Item_Type))
plot(data.frame(table(data.mart.raw$Item_Type)), main="Frequency Distribution of
Item_Type",xlab="Item_Type")
data.frame(table(data.mart.raw$Outlet_Identifier))
plot(data.frame(table(data.mart.raw$Outlet_Identifier)), main="Frequency Distribution of
Outlet_Identifier",xlab="Outlet_Identifier")
data.frame(table(data.mart.raw$Outlet_Size))
plot(data.frame(table(data.mart.raw$Outlet_Size)), main="Frequency Distribution of
Outlet_Size",xlab="Outlet_Size")
data.frame(table(data.mart.raw$Outlet_Location_Type))
plot(data.frame(table(data.mart.raw$Outlet_Location_Type)), main="Frequency Distribution of
Outlet_Location_Type",xlab="Outlet_Location_Type")
data.frame(table(data.mart.raw$Outlet_Type))
plot(data.frame(table(data.mart.raw$Outlet_Type)), main="Frequency Distribution of
Outlet_Type",xlab="Outlet_Type")
data.frame(table(data.mart.raw$Outlet_Establishment_Year))
plot(data.frame(table(data.mart.raw$Outlet_Establishment_Year)), main="Frequency Distribution of
Outlet_Establishment_Year",xlab="Outlet_Establishment_Year")


# Step-5: Exploratory Data Analysis on numerical variables
# After conversion below are numerical variables:
# 1. Item_Weight
# 2. Item_Visibility
# 3. Item_MRP
# 4. Item_Outlet_Sales
summary(data.mart.raw$Item_Weight)
hist(data.mart.raw$Item_Weight)
summary(data.mart.raw$Item_Visibility)
hist(data.mart.raw$Item_Visibility)
summary(data.mart.raw$Item_MRP)
hist(data.mart.raw$Item_MRP)
summary(data.mart.raw$Item_Outlet_Sales)
hist(data.mart.raw$Item_Outlet_Sales)
boxplot(data.mart.raw$Item_Outlet_Sales)


# Step-6: Treating the missing values
# From above exploratoy analysis, we could see there is no normal distriution of data in both factor as well
numerical variable
# So before we normalize them, we need to treat missing values
head(data.mart.raw)
# Treating factor variables
pie(table((data.mart.raw$Item_Fat_Content)),main = "Analysis of Missing Values in Item_Fat_Content")
pie(table((data.mart.raw$Item_Type)),main = "Analysis of Missing Values in Item_Type")
pie(table((data.mart.raw$Outlet_Identifier)),main = "Analysis of Missing Values in Outlet_Identifier")
pie(table((data.mart.raw$Outlet_Establishment_Year)),main = "Analysis of Missing Values in
Outlet_Establishment_Year")
pie(table((data.mart.raw$Outlet_Size)),main = "Analysis of Missing Values in Outlet_Size")
pie(table((data.mart.raw$Outlet_Location_Type)),main = "Analysis of Missing Values in Outlet_Location_Type")
pie(table((data.mart.raw$Outlet_Type)),main = "Analysis of Missing Values in Outlet_Type")
# Treating numerical variables
pie(table(is.na(data.mart.raw$Item_Weight)),main = "Analysis of Missing Values in Item_Weight")
pie(table(is.na(data.mart.raw$Item_Visibility)),main = "Analysis of Missing Values in Item_Visibility")
pie(table(is.na(data.mart.raw$Item_MRP)),main = "Analysis of Missing Values in Item_MRP")
pie(table(is.na(data.mart.raw$Item_Outlet_Sales)),main = "Analysis of Missing Values in Item_Outlet_Sales")
```

```
# Step-6.1: Treating Outlet_Size, Creating split based on the missing values in column Outlet_Size
data.mart.raw.tree <- data.mart.raw
data.mart.raw.tree.test <- data.mart.raw.tree[data.mart.raw.tree$Outlet_Size=="",]
data.mart.raw.tree.train <- data.mart.raw.tree[data.mart.raw.tree$Outlet_Size!="",]

# Step-6.2: Imputing values for outlet_size using decision tree
head(data.mart.raw.tree.train)
#tree_treated <-
rpart(y~age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaign+pdays+previous+poutcome,data=TRAINING_TREATEDBANKPROJECTDATASET)
tree_treated <-
rpart(Outlet_Size~Item_Weight+Item_Fat_Content+Item_Visibility+Item_Type+Item_MRP+Outlet_Identifier+Outlet_Establishment_Year+Outlet_Location_Type+Outlet_Type+Item_Outlet_Sales, data = data.mart.raw.tree.train)
summary(tree_treated)
# Plotting the tree ( it is better though)
plot(tree_treated, uniform=TRUE)
# Now creating the fancy part
fancyRpartPlot(tree_treated)
# We can do prediction as below
predict(tree_treated)
predict(tree_treated, type="class")
# Confusion matrix
table(data.mart.raw.tree.train$Outlet_Size, predict(tree_treated, type="class"), dnn=c("Actual","Predicted"))
# Testing the model with test datpredicted_treated_class1a set
# Loading the file to R
predicted_treated_class <- predict(tree_treated,data.mart.raw.tree.test,type="class")
table(data.mart.raw.tree.test$Outlet_Size,predicted_treated_class,dnn=c("Actual","Predicted"))
# treating the missing values
for (i in 1 : length(data.mart.raw.tree.test$Outlet_Size)){
   if(data.mart.raw.tree.test$Outlet_Identifier[i] == ("OUT018") |
     data.mart.raw.tree.test$Outlet_Identifier[i] == ("OUT027") |
     data.mart.raw.tree.test$Outlet_Identifier[i] == ("OUT049")){
      data.mart.raw.tree.test$Outlet_Size[i] <- as.character("Medium")
   } else if (data.mart.raw.tree.test$Outlet_Identifier[i] == ("OUT013")){
      data.mart.raw.tree.test$Outlet_Size[i] <- as.character("High")
   } else {data.mart.raw.tree.test$Outlet_Size[i] <- as.character("Small")}
}
tail(data.mart.raw.tree.test$Outlet_Size)
data.mart.raw.tree <- rbind(data.mart.raw.tree.train,data.mart.raw.tree.test)
tail(data.mart.raw.tree)
data.mart.raw.2 <- data.mart.raw.tree

# Step:6.3 Treating Item_Weight
data.mart.raw.3 <- data.mart.raw.2
tail(data.mart.raw.3)
summary(data.mart.raw.3$Item_Weight) # <- from summary we see mean and median stay close, so i will fill data with its mean value
for (i in 1 : length(data.mart.raw.3$Item_Weight)){
   if(is.na(data.mart.raw.3$Item_Weight[i]) == TRUE |
     is.nan(data.mart.raw.3$Item_Weight[i]) == TRUE |
     is.null(data.mart.raw.3$Item_Weight[i]) == TRUE){
      data.mart.raw.3$Item_Weight[i] <- mean(data.mart.raw.3$Item_Weight, na.rm = TRUE)
   }
}
summary(data.mart.raw.3$Item_Weight) # <- From this we could see that mean and median became so close and hence we can hope this imputation works fine
data.mart.treaded <- data.mart.raw.3
```

```
hist(data.mart.treaded$Item_Weight) #<- Converted from normal curve

# Step:6.4 Treating Item_Weight Item_Fat_Content
data.frame(table(data.mart.treaded$Item_Fat_Content))
plot(data.frame(table(data.mart.treaded$Item_Fat_Content)), main="Frequency Distribution of
Item_Fat_Content",xlab="Item_Fat_Content")
data.mart.treaded$Item_Fat_Content <- as.character(data.mart.treaded$Item_Fat_Content)
for (i in 1 : length(data.mart.treaded$Item_Fat_Content)){
   if(data.mart.treaded$Item_Fat_Content[i] == as.character("LF") |
      data.mart.treaded$Item_Fat_Content[i] == as.character("low fat") |
      data.mart.treaded$Item_Fat_Content[i] == as.character("Low Fat")){
       data.mart.treaded$Item_Fat_Content[i] <- as.character("Low_Fat")
   } else {data.mart.treaded$Item_Fat_Content[i] <- as.character("Regular")}
}

# Step:6.5 Converting the Column objects to factor or Numeric after treatment
data.mart.treaded$Item_Fat_Content <- as.factor(data.mart.treaded$Item_Fat_Content)
data.mart.treaded$Outlet_Size <- factor(data.mart.treaded$Outlet_Size,levels=c("High", "Medium", "Small"))

# Step:7 Splitting the dataset to test and train for local validation
# Creating a random index to split the data as 80 - 20%
idx <- createDataPartition(data.mart.treaded$Item_Weight, p=.80, list=FALSE)
print(idx[1:20])
# Using the index created to create a Training Data set - 131 observations created
data.train <- data.mart.treaded[idx,]
head(data.mart.treaded)
# Using the index created to create a Testing Data set - 31 observations created
data.test <- data.mart.treaded[-idx,]
head(data.test)
idx <- NULL

# Step-8 Exploratory data analysis on training set
# Factor Variables
data.frame(table(data.train$Item_Fat_Content))
plot(data.frame(table(data.train$Item_Fat_Content)), main="Frequency Distribution of
Item_Fat_Content",xlab="Item_Fat_Content")
data.frame(table(data.train$Item_Type))
plot(data.frame(table(data.train$Item_Type)), main="Frequency Distribution of Item_Type",xlab="Item_Type")
data.frame(table(data.train$Outlet_Identifier))
plot(data.frame(table(data.train$Outlet_Identifier)), main="Frequency Distribution of
Outlet_Identifier",xlab="Outlet_Identifier")
data.frame(table(data.train$Outlet_Size))
plot(data.frame(table(data.train$Outlet_Size)), main="Frequency Distribution of Outlet_Size",xlab="Outlet_Size")
data.frame(table(data.train$Outlet_Location_Type))
plot(data.frame(table(data.train$Outlet_Location_Type)), main="Frequency Distribution of
Outlet_Location_Type",xlab="Outlet_Location_Type")
data.frame(table(data.train$Outlet_Type))
plot(data.frame(table(data.train$Outlet_Type)), main="Frequency Distribution of
Outlet_Type",xlab="Outlet_Type")
data.frame(table(data.train$Outlet_Establishment_Year))
plot(data.frame(table(data.train$Outlet_Establishment_Year)), main="Frequency Distribution of
Outlet_Establishment_Year",xlab="Outlet_Establishment_Year")
# Numerical Variabes
summary(data.train$Item_Weight)
hist(data.train$Item_Weight)
summary(data.train$Item_Visibility)
hist(data.train$Item_Visibility)
```

```
summary(data.train$Item_MRP)
hist(data.train$Item_MRP)
summary(data.train$Item_Outlet_Sales)
hist(data.train$Item_Outlet_Sales)
pie(table((data.train$Outlet_Size)),main = "Analysis of Missing Values in Outlet_Size")
pie(table(is.na(data.train$Item_Weight)),main = "Analysis of Missing Values in Item_Weight")

# Step-9 : Making Inference and Hypothesis
# 1. Low fat food is being purchased more compare to the regular fat foods
# 2. Food products like Fruits and Vegitables, snaks have higher sale; Households, canned, dairy and baking good have average sales and others are bought even less
# 3. OUT010 and OUT019 have lowest sale compare to others
# 4. Big mart owns Small and medium sized outlets more when comapre to High size outlet
# 5. Big mart outlets are situated more more in Tier3 and Tier2 locations when compare to Tier1 regions
# 6. Other than 1997, we could see a constant sale obtained in all years till
# 7. Item weight has a normal distribution, which means product of all weight are available in store at equal proportion, it not just the whole sale which is happening in store
# 8. Product visibility is sckewed to right, stores have more of small display area for product more and interestingly there is a size 0 which can be even online sold product
# 9. MRP of the product is also quite normally distributed, which means product of all price range from $31 to $266 is available in store in eqal proportion, so it target all kind of customers for its sales
# 10. Total sale revenue is skewed to right, meaning store constantly generate revenue of range $800 to $3000 in each of its outlet mostly
# Hypothesis: Groceries like fruit, vegetables and snkacks with low fat content with minimum product visibility in a small and medium sized outlet situated in Tire-3 and Tier-2 region should have a comparitively good sale excluding the outlets OUT010 and OUT019.

# Step-10 : Basic Model Building
model1 <-
lm(Item_Outlet_Sales~Item_Fat_Content+Item_Type+Outlet_Identifier+Outlet_Establishment_Year+Outlet_Size+Outlet_Location_Type+Outlet_Type+Item_Weight+Item_Visibility+Item_MRP,data = data.train)
cor_var1 <- data.frame(data.train$Item_Weight,data.train$Item_Visibility,data.train$Item_MRP)
cor(cor_var1) # No significant correlation exists with all numerical variabels available
summary(model1) # <- model-1 explains 0.5657 of sales variance, having Item_Fat_Content, Outlet_Identifier and Item_MRP as a significant variables
# Item_Outlet_Sales ~ Item_Fat_Content + Outlet_Identifier + Item_MRP

# Step-11 : Model Building using stepwise algorithm
model2_stepwise <- step(model1, direction = "backward")
summary(model2_stepwise) # <- explains 0.566 of sales variance
# Item_Outlet_Sales ~ Outlet_Identifier + Item_MRP

# Step-12: Residual Analysis
par(mfrow=c(4,2))
par(mar = rep(2, 4))
plot(model2_stepwise)
sd(data.train$Item_Outlet_Sales)
residual <- rstandard(model2_stepwise)
hist(residual) # Residual seems normally distributed
# Could observe some heteroscadastic behavious in residual plot, we can try for some transformation

# Step-13: Transformation
# Doing log transformation on dependent variable
model3_transformed <-
lm(log(Item_Outlet_Sales)~Item_Fat_Content+Item_Type+Outlet_Identifier+Outlet_Establishment_Year+Outlet_Size+Outlet_Location_Type+Outlet_Type+Item_Weight+Item_Visibility+Item_MRP,data = data.train)
summary(model3_transformed)
```

```
# Adj R^2 is 0.7241
par(mfrow=c(4,2))
par(mar = rep(2, 4))
plot(model3_transformed)
residual_af_tranformation <- rstandard(model3_transformed)
hist(residual_af_tranformation)

# Step-14: Outlier Check and Influential Point Check
# computing studentized residual for outlier check
n_sample_size <- nrow(data.train)
studentized.residuals <- studres(model3_transformed)
#cat("Complete list of Studentized Residual::::","\n")
#print(studentized.residuals)
for(i in c(1:n_sample_size)){
   if(studentized.residuals[i] < -3 || studentized.residuals[i] > 3){
      cat("Validate these values for outliers:::",studentized.residuals[i],"at observation",i,"\n")
   }
}
# Influential Points
hhat.model <- lm.influence(model3_transformed)$hat
n_sample_size <- nrow(data.train)
p_beta <- length(model3_transformed$coefficients) +1
#cat("Complete list of HHat Values::::","\n")
#print(hhat.model)
hhat.cutoff <- (2*p_beta)/n_sample_size
cat("Looking for values more than cut off::::",hhat.cutoff,"\n")
for(i in c(1:n_sample_size)){
   if(hhat.model[i] > hhat.cutoff){
      cat("Validate these values for Influential points:::",hhat.model[i],"at observation",i,"\n")
   }
}
# we see only observation 831 as both outlier and influential point, so trying to remove it
data.train.treated <- data.train[-c(831),]
model3_transformed_treated <-
lm(log(Item_Outlet_Sales)~Item_Fat_Content+Item_Type+Outlet_Identifier+Outlet_Establishment_Year+Outlet_Size+Outlet_Location_Type+Outlet_Type+Item_Weight+Item_Visibility+Item_MRP,data = data.train.treated)
summary(model3_transformed_treated)
# removing the outlier impoves the Adj R-square very significantly

# Ste-15: Model validation for Multicollinearity
# vif(model3_transformed) # No aliased coefficient in the model

# Step-16: Computing the standardized coefficient
#data.train.std <- sapply(data.train[,],FUN=scale)
#data.train.std <- data.frame(data.train)
#model3_transformed.std <-
lm(log(Item_Outlet_Sales)~Item_Fat_Content+Item_Type+Outlet_Identifier+Outlet_Establishment_Year+Outlet_Size+Outlet_Location_Type+Outlet_Type+Item_Weight+Item_Visibility+Item_MRP, data = data.train)
#summary(model3_transformed.std)
#since most of the variables are factorial in nature, there is no need of standardizing the value

# Step-17: Model Validation
FINAL_MODEL <- lm(log(Item_Outlet_Sales) ~ Outlet_Identifier + Item_MRP, data = data.train)
final_summary <- summary(FINAL_MODEL); final_summary # adj r-square is 72.41%
str(data.test)
COUNT_PREDICTED <- predict(FINAL_MODEL,data.test)
plot(COUNT_PREDICTED,data.test$Item_Outlet_Sales,lwd=2, cex=2, col="red")
```

```
COUNT_PREDICTED_RE_TRANSFORMED <- exp(COUNT_PREDICTED)
plot(COUNT_PREDICTED_RE_TRANSFORMED,data.test$count,lwd=2, cex=2, col="green")
abline(0,1,col='red', lwd=2)

# Step-18: Prediction
# Prediction Interval
pred_Int <- predict(FINAL_MODEL,data.test,interval = "predict")
conf_Int <- predict(FINAL_MODEL,data.test,interval = "confidence")
converted_pred_int <- exp(pred_Int)
converted_conf_int <- exp(conf_Int)
data.test$predicted_count <- converted_pred_int[,1]
data.test$prediction_interval_low <- converted_pred_int[,2]
data.test$prediction_interval_high <- converted_pred_int[,3]
data.test$confidence_interval_low <- converted_conf_int[,2]
data.test$confidence_interval_high <- converted_conf_int[,3]
data.prediction.result <-
data.frame(data.test$Item_Outlet_Sales,data.test$predicted_count,data.test$prediction_interval_low,data.test$
prediction_interval_high,data.test$confidence_interval_low,data.test$confidence_interval_high)
View(data.prediction.result)
data.test$predicted_count <- NULL
data.test$prediction_interval_low <- NULL
data.test$prediction_interval_high <- NULL
data.test$confidence_interval_low <- NULL
data.test$confidence_interval_high <- NULL
```