Sales Analysis - Mid Report

Mughundhan

9/9/2017

1. About the Project

The dataset comprises of sales data (of a renowned Super Market) for 1559 products across 10 stores in different cities (broadly classified based on the purchase power parity, working population, size and few other factors).

The project aims to build a predictive model to analyze the sales of each product at a particular store. With this we shall understand the properties of products and stores which play a key role in increasing sales. The results of the model will be used to provide recommendations to improve the sales.

1.1. NOTES

- To evaluate how good is a model, let us understand the impact of wrong predictions. If we predict sales to be higher than what they might be, the store will spend a lot of money making unnecessary arrangement which would lead to excess inventory. On the other side if I predict it too low, I will lose out on sales opportunity.

2. Creating an appropriate Environment

```
rm(list = ls())
setwd('/Users/Mughundhan/UIC/UIC Academics/FALL 2017/BIZ ANALYTICS STATS/Proj
ect/Mid Report')
library(lubridate) # for csv files
library(leaflet)   # interactive maps
library(dplyr)     # for piping purpose %>%
library(data.table)# aggregate
library(ggplot2)   # barplot
library(mice)      # imputing with plausible data values (drawn from a distri
bution specifically designed for each missing datapoint)
library(rpart)     # Decision Trees
library(VIM)       # Visual Representation for MICE
library(data.table)
train <- read.csv("Train.csv", header=T, na.strings=c("","NA")) #Empty spaces
to be replaced by NA
test <- read.csv("Test.csv", header=T, na.strings=c("","NA"))
test$Item_Outlet_Sales <- NA
fdata <- rbind(test, train)
fdata <- as.data.table(fdata)
```

3. Data Exploration

3.1 Data Dictionary

Let us have a look at the description of each variable in the dataset:

1. **Item_Identifier**: Unique Product ID
2. **Item_Weight**: Weight of the Product
3. **Item_Fat_Content**: How much fat content the product contains (Low, Regular)
4. **Item_Visibility**: The percent of *total display area* of all products in a store allocated to the particular product
5. **Item_Type**: The Category to which the product belongs (eg: Breakfast, Soft Drinks, Household etc)
6. **Item_MRP**: Maximum Retail Price of the Product (Indian Rupees)
7. **Outlet_Identifier**: Unique Store ID - multiple stores located at different cities
8. **Outlet_Establishment_Year**: The year, when the store started its operation
9. **Outlet_Size**: Size of the store (High, Medium, Small)
10. **Outlet_Location_Type**: The type of the city in which the store is located (Tier1, Tier2 ....)
11. **Outlet_Type**: The type of the outlet (Grocery store or a Super Market)
12. **Item_Outlet_Sales**: Sales of the product in the particular store. [*Outcome Variable to be predicted*]

3.2 Overview of the dataset with R

Let us now perform basic operations to have a look at the summary and the structure of the dataset.

```
summary(fdata)

##  Item_Identifier  Item_Weight     Item_Fat_Content Item_Visibility
##  DRA24   :   10   Min.   : 4.555   LF      : 522     Min.   :0.00000
##  DRA59   :   10   1st Qu.: 8.710   Low Fat:8485     1st Qu.:0.02704
##  DRB25   :   10   Median :12.600   Regular:4824     Median :0.05402
##  DRC25   :   10   Mean   :12.793   low fat: 178     Mean   :0.06595
##  DRC27   :   10   3rd Qu.:16.750   reg     : 195     3rd Qu.:0.09404
##  DRC36   :   10   Max.   :21.350                     Max.   :0.32839
##  (Other):14144   NA's   :2439
##                 Item_Type        Item_MRP      Outlet_Identifier
##  Fruits and Vegetables:2013   Min.   : 31.29   OUT027 :1559
##  Snack Foods          :1989   1st Qu.: 94.01   OUT013 :1553
##  Household            :1548   Median :142.25   OUT035 :1550
##  Frozen Foods         :1426   Mean   :141.00   OUT046 :1550
##  Dairy                :1136   3rd Qu.:185.86   OUT049 :1550
##  Baking Goods         :1086   Max.   :266.89   OUT045 :1548
##  (Other)              :5006                    (Other):4894
##  Outlet_Establishment_Year Outlet_Size   Outlet_Location_Type
##  Min.   :1985              High  :1553   Tier 1:3980
```

```
##  1st Qu.:1987             Medium:4655   Tier 2:4641
##  Median :1999             Small :3980   Tier 3:5583
##  Mean   :1998             NA's  :4016
##  3rd Qu.:2004
##  Max.   :2009
##
##             Outlet_Type   Item_Outlet_Sales
##  Grocery Store    :1805   Min.   :   33.29
##  Supermarket Type1:9294   1st Qu.:  834.25
##  Supermarket Type2:1546   Median : 1794.33
##  Supermarket Type3:1559   Mean   : 2181.29
##                           3rd Qu.: 3101.30
##                           Max.   :13086.97
##                           NA's   :5681
```

```
str(fdata)
```

```
## Classes 'data.table' and 'data.frame':   14204 obs. of  12 variables:
##  $ Item_Identifier          : Factor w/ 1559 levels "DRA12","DRA24",..: 11
04 1068 1407 810 1185 462 605 267 669 171 ...
##  $ Item_Weight              : num  20.75 8.3 14.6 7.32 NA ...
##  $ Item_Fat_Content         : Factor w/ 5 levels "LF","Low Fat",..: 2 5 2
2 3 3 3 2 3 2 ...
##  $ Item_Visibility          : num  0.00756 0.03843 0.09957 0.01539 0.1186
...
##  $ Item_Type                : Factor w/ 16 levels "Baking Goods",..: 14 5
12 14 5 7 1 1 14 1 ...
##  $ Item_MRP                 : num  107.9 87.3 241.8 155 234.2 ...
##  $ Outlet_Identifier        : Factor w/ 10 levels "OUT010","OUT013",..: 10
3 1 3 6 9 4 6 8 3 ...
##  $ Outlet_Establishment_Year: int  1999 2007 1998 2007 1985 1997 2009 1985
2002 2007 ...
##  $ Outlet_Size              : Factor w/ 3 levels "High","Medium",..: 2 NA
NA NA 2 3 2 2 NA NA ...
##  $ Outlet_Location_Type     : Factor w/ 3 levels "Tier 1","Tier 2",..: 1 2
3 2 3 1 3 3 2 2 ...
##  $ Outlet_Type              : Factor w/ 4 levels "Grocery Store",..: 2 2 1
2 4 2 3 4 2 2 ...
##  $ Item_Outlet_Sales        : num  NA NA NA NA NA NA NA NA NA NA ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Observation

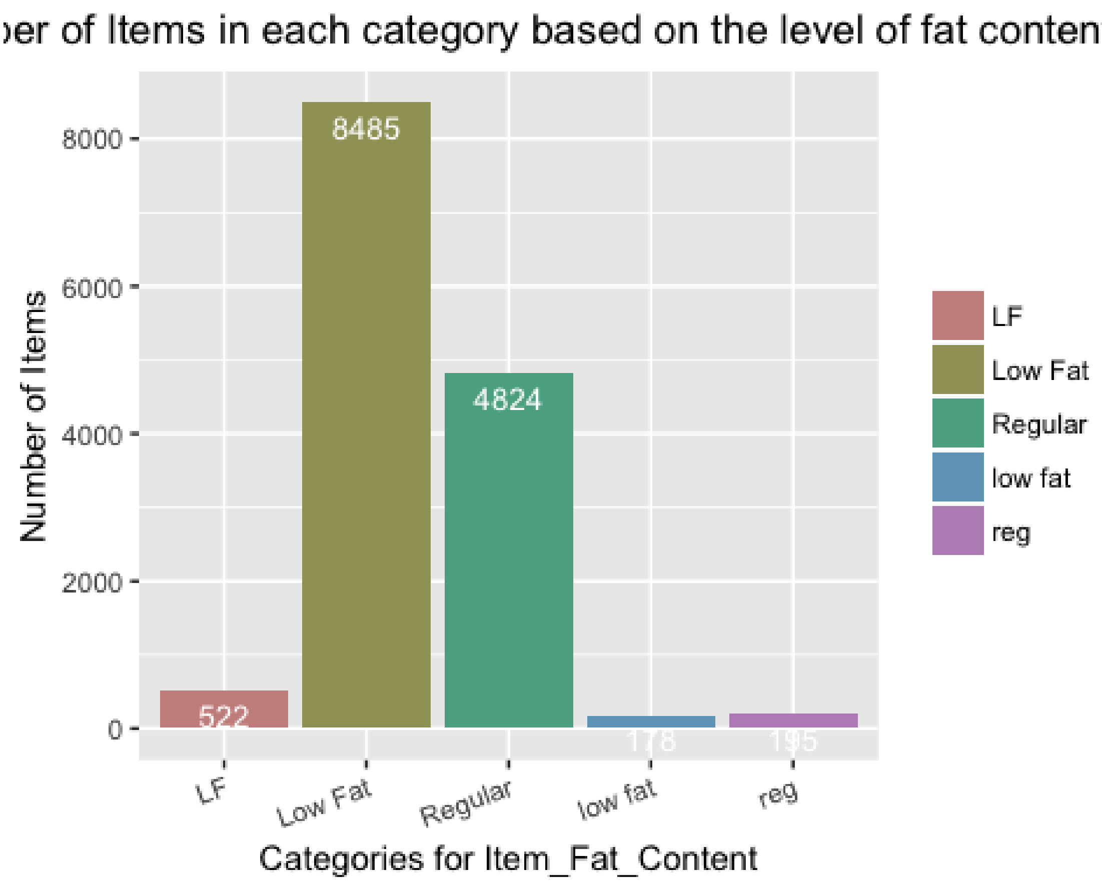
1. There are 11 + 1 variables in the dataset (1-target variable: Item_Outlet_Sales)
2. We shall perform number operations on 3 numerical variables: *Item_Weight, Item_Visibility, Item_MRP*
3. There are several factor variables which will be transformed into character variables for feature engineering purpose: *Item_Fat_Content, Outlet_Identifier, Outlet_Size, Outlet_Location_Type, Outlet_Type*

4. There is only one variable with information regarding the date: *Outlet_Establishment_Year*. We might perform simple numerical operations since only the year is given.
5. Few variables (*Outlet_Size, Item_Weight*) contain missing values which needs to be imputed.

3.2 Deeper Insights from the dataset using R functions

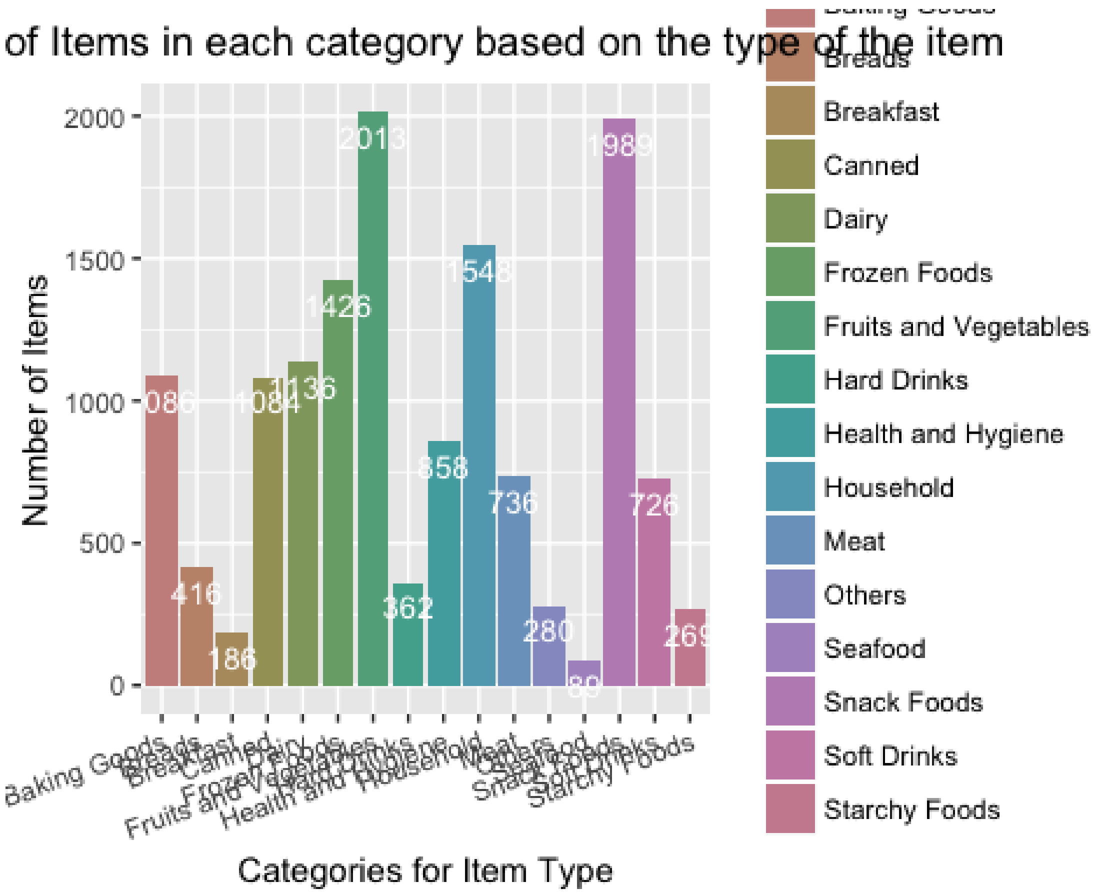
```
sapply(fdata, function(x) length(unique(x))) #Number of Unique Values in each
column

##           Item_Identifier               Item_Weight
##                      1559                       416
##          Item_Fat_Content           Item_Visibility
##                         5                     13006
##                 Item_Type                  Item_MRP
##                        16                      8052
##         Outlet_Identifier Outlet_Establishment_Year
##                        10                         9
##               Outlet_Size      Outlet_Location_Type
##                         4                         3
##               Outlet_Type         Item_Outlet_Sales
##                         4                      3494

sapply(fdata, function(x) sum(is.na(x))) #Number of Missing Values in each co
lumn

##           Item_Identifier               Item_Weight
##                         0                      2439
##          Item_Fat_Content           Item_Visibility
##                         0                         0
##                 Item_Type                  Item_MRP
##                         0                         0
##         Outlet_Identifier Outlet_Establishment_Year
##                         0                         0
##               Outlet_Size      Outlet_Location_Type
##                      4016                         0
##               Outlet_Type         Item_Outlet_Sales
##                         0                      5681

table(fdata$Item_Fat_Content) #Frequency of categories for Item_Fat_Content

##
##      LF Low Fat Regular low fat     reg
##     522    8485    4824     178     195

ggplot(fdata, aes(x=as.factor(Item_Fat_Content), fill=as.factor(Item_Fat_Cont
ent) )) +
  geom_bar() +
  stat_count(aes(label = ..count..), geom = "text", vjust=1.6, size=3.5, colo
r="white") +
  scale_fill_hue(c = 40) +
```

```
  labs(x="Categories for Item_Fat_Content", y="Number of Items", title="Numbe
r of Items in each category based on the level of fat content") +
  theme(legend.title=element_blank(), plot.title = element_text(hjust = 0.5))
+
  theme(axis.text.x = element_text(angle = 20, hjust = 1))
```




```
table(fdata$Item_Type) #Frequency of categories for Item_Type

##
##          Baking Goods                Breads             Breakfast
##                  1086                   416                   186
##                Canned                 Dairy         Frozen Foods
##                  1084                  1136                  1426
## Fruits and Vegetables           Hard Drinks    Health and Hygiene
##                  2013                   362                   858
##             Household                  Meat                Others
##                  1548                   736                   280
##               Seafood           Snack Foods           Soft Drinks
##                    89                  1989                   726
##         Starchy Foods
##                   269

ggplot(fdata, aes(x=as.factor(Item_Type), fill=as.factor(Item_Type) )) +
  geom_bar() +
```

```r
  stat_count(aes(label = ..count..), geom = "text", vjust=1.6, size=3.5, colo
r="white") +
  scale_fill_hue(c = 40) +
  labs(x="Categories for Item Type", y="Number of Items", title="Number of It
ems in each category based on the type of the item") +
  theme(legend.title=element_blank(), plot.title = element_text(hjust = 0.5))
+
  theme(axis.text.x = element_text(angle = 20, hjust = 1))
```

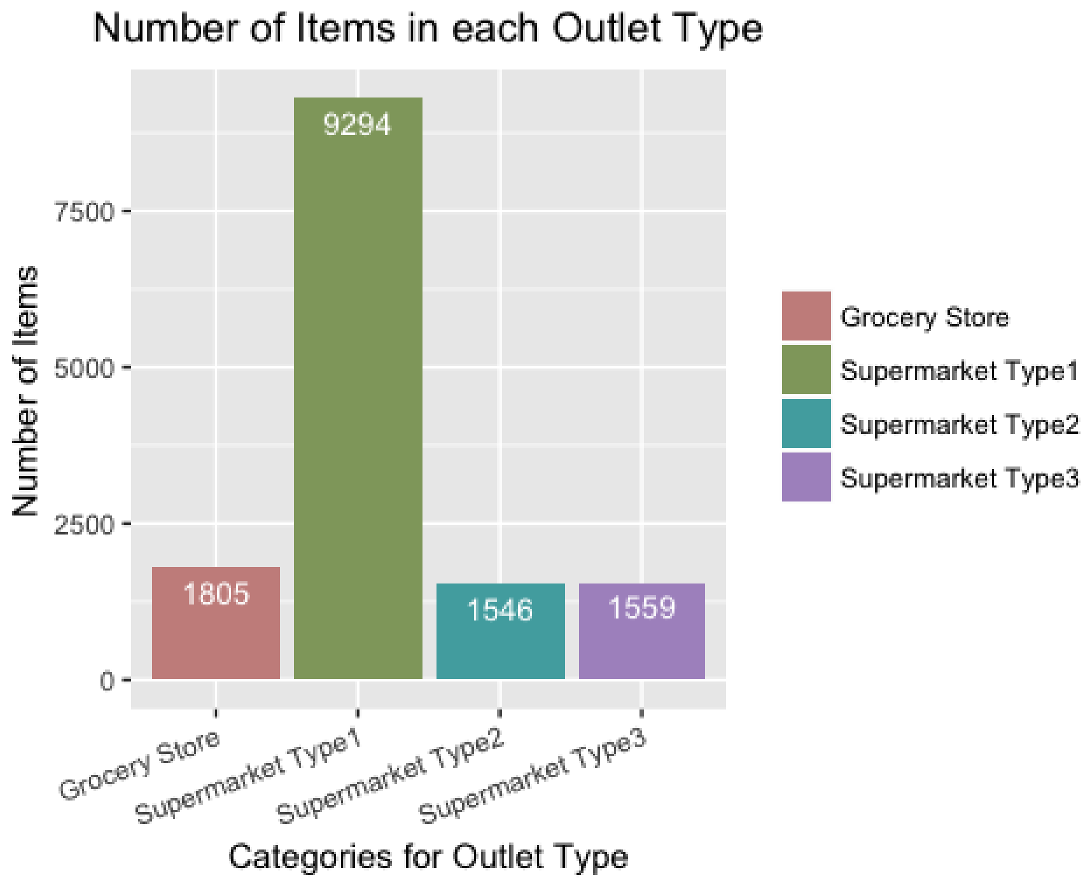



```r
table(fdata$Outlet_Location_Type) #Frequency of categories for Outlet_Locatio
n_Type
```

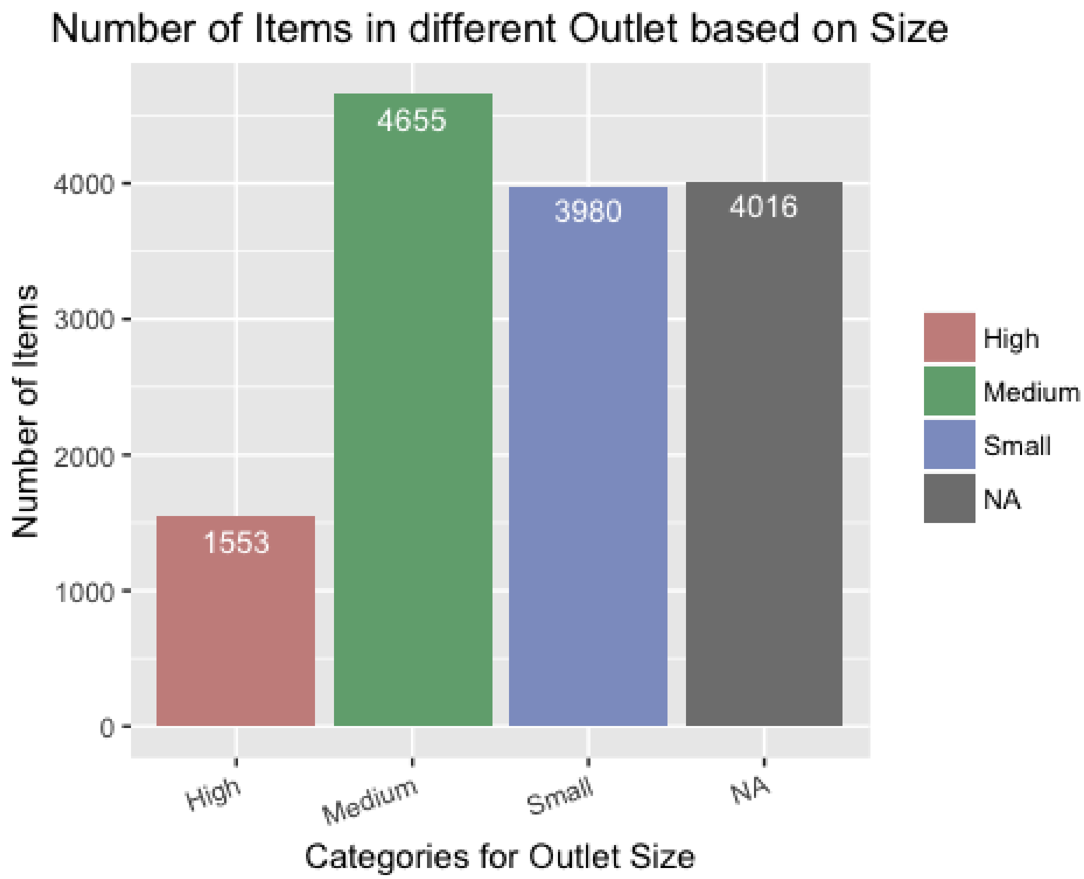
```
##
## Tier 1 Tier 2 Tier 3
##   3980   4641   5583
```

```r
ggplot(fdata, aes(x=as.factor(Outlet_Type), fill=as.factor(Outlet_Type) )) +
  geom_bar() +
  stat_count(aes(label = ..count..), geom = "text", vjust=1.6, size=3.5, colo
r="white") +
  scale_fill_hue(c = 40) +
  labs(x="Categories for Outlet Type", y="Number of Items", title="Number of
Items in each Outlet Type") +
  theme(legend.title=element_blank(), plot.title = element_text(hjust = 0.5))
```

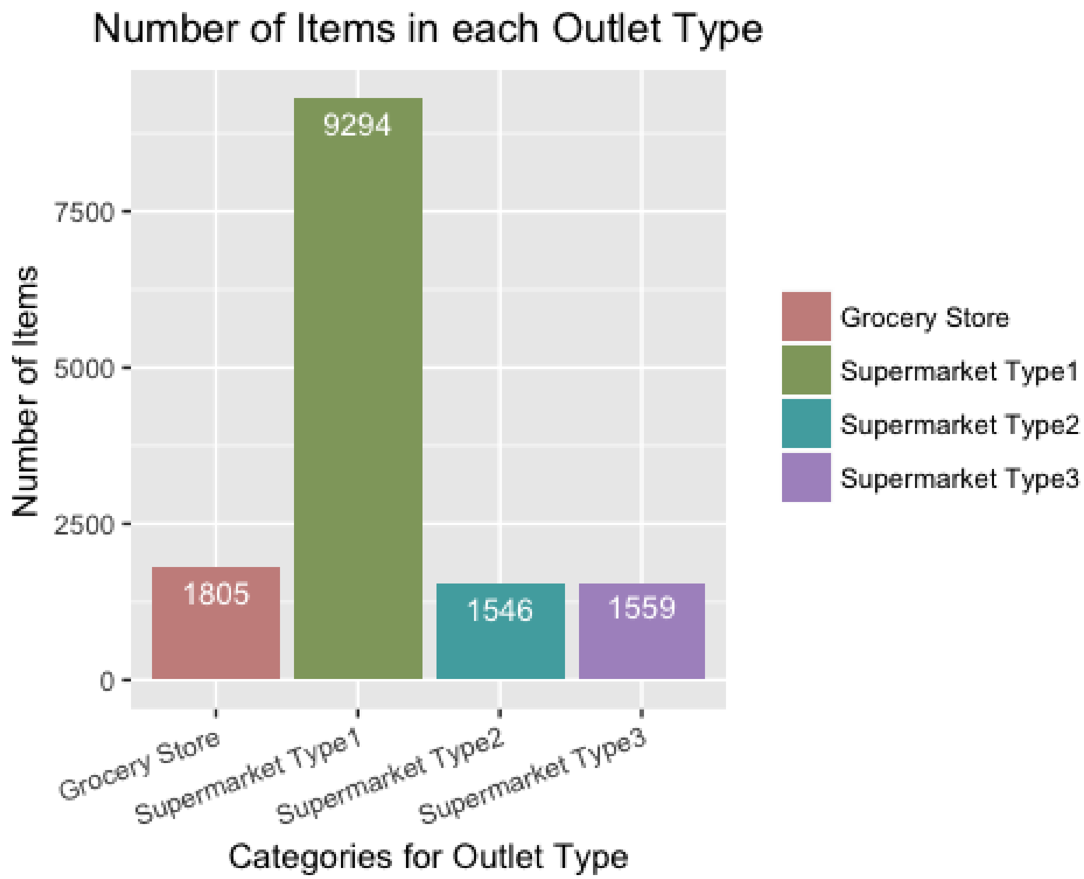
```
+
  theme(axis.text.x = element_text(angle = 20, hjust = 1))
```




```
table(fdata$Outlet_Size) #Frequency of categories for Outlet_Size

##
##   High Medium  Small
##   1553   4655   3980

ggplot(fdata, aes(x=as.factor(Outlet_Size), fill=as.factor(Outlet_Size) )) +
  geom_bar() +
  stat_count(aes(label = ..count..), geom = "text", vjust=1.6, size=3.5, colo
r="white") +
  scale_fill_hue(c = 40) +
  labs(x="Categories for Outlet Size", y="Number of Items", title="Number of
Items in different Outlet based on Size") +
  theme(legend.title=element_blank(), plot.title = element_text(hjust = 0.5))
+
  theme(axis.text.x = element_text(angle = 20, hjust = 1))
```




```
table(fdata$Outlet_Type) #Frequency of categories for Outlet_Type

##
##     Grocery Store Supermarket Type1 Supermarket Type2 Supermarket Type3
##              1805              9294              1546              1559

ggplot(fdata, aes(x=as.factor(Outlet_Type), fill=as.factor(Outlet_Type) )) +
  geom_bar() +
  stat_count(aes(label = ..count..), geom = "text", vjust=1.6, size=3.5, colo
r="white") +
  scale_fill_hue(c = 40) +
  labs(x="Categories for Outlet Type", y="Number of Items", title="Number of
Items in each Outlet Type") +
  theme(legend.title=element_blank(), plot.title = element_text(hjust = 0.5))
+
  theme(axis.text.x = element_text(angle = 20, hjust = 1))
```



Observation: 1. We can observe the number of missing values and the number of unique values (levels) in each column using sapply. 2. The graphs display the distribution and contribution of each sub-category corresponding to that variable.

4. Hypotheses Generation

Based on the basic data exploration, we shall have two levels of hypotheses: **1. Store-level; 2. Product-level**. Both plays a crucial role in determining the sales of each product at specific stores located across different cities. The hypotheses generated at both the levels based on the available dataset are as follows:

I. Product-Level Hypotheses

1. Item_Fat_Content: Items are classified based on the fat content. Since we consume on low fat items as a part of our regular diet, It is highly possible that ***Low fat*** items are generally sold more than the items with high fat content.

2. Item_Type: Items which we use on ***regular basis*** - like ready to eat, soft drinks has higher probability of being sold when compared with luxury items.

3. Item_MRP: More expensive items might be bought occasionally. Items with ***lower prices*** might be a product which is being used on a regular basis. Thus, Low priced items might have sold better than expensive items.

II. Store-Level Hypotheses

1. Outlet_Size: ***Bigger outlets*** might attract bigger crowds. This results in increasing the sales of the products in that specific store.

2. Outlet_Location_Type: ***Bigger cities*** or cities with high population density has a larger customer base for the stores at their location. Stores located in Tier-1 cities might have better sales than stores located in other types of cities.

3. Outlet_Type: Similar to the previous hypotheses. ***Supermarkets*** look more fancy than grocery shops. Among supermarket, the highest among this sub-classification might attract larger crowds and emerge as the best selling store when compared with other outlet types.

5. Handling Missing Values

5.1 Finding the missing values

Identifying the missing values column-wise. The name of the column and the corresponding number of missing values in each column is given.

5.2 Imputing the missing values

1. Item_Weight and Item_Identifier: Taking average of Item_Weight based on Item_Identifier and imputing missing values in Item_Weight

```
length(unique(fdata$Item_Identifier)) #Identify no. of unique values in the I
tem_Identifier attribute

## [1] 1559

avg_Item_Weight <- aggregate(Item_Weight~Item_Identifier, data=fdata, FUN=fun
ction(x) c(mean=mean(x), count=length(x))) #making an aggregate - similar to
group by feature in SQL
avg_Item_Weight <- as.data.table(avg_Item_Weight) #converting into data.table
for easier computation

cdata <- merge(fdata, avg_Item_Weight, by="Item_Identifier") #merging the dat
a

for(i in 1:nrow(cdata))
{
  if(is.na(cdata[i,2]))
  {
    cdata$Item_Weight.x[i] <- cdata$Item_Weight.y[i] #missing weights replace
d by average weight of the item depending on the unique Item_Identifier
  }
}

fdata <- cdata[ ,1:(ncol(cdata)-1)] #deleting the unnecessary column created
during the imputation process
```

```r
#View(cdata)
names(fdata)[names(fdata)=="Item_Weight.x"] <- "Item_Weight" #Renaming the at
tribute
sapply(fdata, function(x) sum(is.na(x))) #Number of Missing Values in each co
lumn

##           Item_Identifier                Item_Weight
##                         0                          0
##          Item_Fat_Content            Item_Visibility
##                         0                          0
##                 Item_Type                   Item_MRP
##                         0                          0
##         Outlet_Identifier Outlet_Establishment_Year
##                         0                          0
##               Outlet_Size       Outlet_Location_Type
##                      4016                          0
##               Outlet_Type          Item_Outlet_Sales
##                         0                       5681

#View(fdata)

rm(cdata, i)
```

2. Outlet_Size and Outlet_Type: Taking average of Outlet_Size based on Outlet_Type and imputing missing values in Outlet_Size

```r
table(fdata$Outlet_Type, fdata$Outlet_Size)

##
##                     High Medium Small
##   Grocery Store        0      0   880
##   Supermarket Type1 1553   1550  3100
##   Supermarket Type2    0   1546     0
##   Supermarket Type3    0   1559     0

round(prop.table(table(fdata$Outlet_Type, fdata$Outlet_Size), 1), 2) #Identif
y the proportion

##
##                     High Medium Small
##   Grocery Store     0.00   0.00  1.00
##   Supermarket Type1 0.25   0.25  0.50
##   Supermarket Type2 0.00   1.00  0.00
##   Supermarket Type3 0.00   1.00  0.00
```

Observation:

1. All Grocery Store -> Small
2. Most Super Market 1 -> Small
3. All Super Market 2 -> Medium
4. All Super Market 3 -> Medium

```r
fdata$Outlet_Size[is.na(fdata$Outlet_Size) & fdata$Outlet_Type == "Grocery St
ore"] <- "Small"
fdata$Outlet_Size[is.na(fdata$Outlet_Size) & fdata$Outlet_Type == "Supermarke
t Type1"] <- "Small"
fdata$Outlet_Size[is.na(fdata$Outlet_Size) & fdata$Outlet_Type == "Supermarke
t Type2"] <- "Medium"
fdata$Outlet_Size[is.na(fdata$Outlet_Size) & fdata$Outlet_Type == "Supermarke
t Type3"] <- "Medium"
sapply(fdata, function(x) sum(is.na(x))) #Number of Missing Values in each co
lumn
```

```
##           Item_Identifier               Item_Weight
##                         0                         0
##          Item_Fat_Content           Item_Visibility
##                         0                         0
##                 Item_Type                  Item_MRP
##                         0                         0
##         Outlet_Identifier Outlet_Establishment_Year
##                         0                         0
##               Outlet_Size      Outlet_Location_Type
##                         0                         0
##               Outlet_Type         Item_Outlet_Sales
##                         0                      5681
```

```r
table(fdata$Outlet_Type, fdata$Outlet_Size)
```

```
## 
##                      High Medium Small
##   Grocery Store         0      0  1805
##   Supermarket Type1  1553   1550  6191
##   Supermarket Type2     0   1546     0
##   Supermarket Type3     0   1559     0
```

```r
round(prop.table(table(fdata$Outlet_Type, fdata$Outlet_Size), 1), 2)
```

```
## 
##                      High Medium Small
##   Grocery Store      0.00   0.00  1.00
##   Supermarket Type1  0.17   0.17  0.67
##   Supermarket Type2  0.00   1.00  0.00
##   Supermarket Type3  0.00   1.00  0.00
```

6. Feature Engineering

We explored some nuances in the data in the data exploration section. Now let us try to resolve them and make our data ready for analysis. We will also create some new variables using the existing ones in this section.

6.1. Consider combining Outlet_Type

During exploration, we decided to consider combining the Supermarket Type2 and Type3 variables. But is that a good idea? A quick way to check that could be to analyze the mean sales by type of store. If they have similar sales, then keeping them separate won???t help much.

```
avg_Item_Sales <- aggregate(Item_Outlet_Sales~Outlet_Type, data=fdata, FUN=fu
nction(x) c(mean=mean(x), count=length(x)))
avg_Item_Sales <- as.data.table(avg_Item_Sales)
rm(avg_Item_Sales)
```

Observation This shows significant difference between Supermarket Type2 and Type3 variables, hence we???ll leave them as it is.

6.2. Modify Item_Visibility

We noticed that the minimum value here is 0, which makes no practical sense. Lets consider it like missing information and impute it with mean visibility of that product.

```
summary(fdata$Item_Visibility)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.00000 0.02704 0.05402 0.06595 0.09404 0.32839

rm(cdata)
length(unique(fdata$Item_Identifier))

## [1] 1559

avg_Item_Visibility <- aggregate(Item_Visibility~Item_Identifier, data=fdata,
FUN=function(x) c(mean=mean(x), count=length(x)))
avg_Item_Visibility <- as.data.table(avg_Item_Visibility)

cdata <- merge(fdata, avg_Item_Visibility, by="Item_Identifier")

for(i in 1:nrow(cdata))
{
  if(cdata[i,4]==0)
  {
    cdata$Item_Visibility.x[i] <- cdata$Item_Visibility.y[i]
  }
}

fdata <- cdata[ ,1:(ncol(cdata)-1)]
names(fdata)[names(fdata)=="Item_Visibility.x"] <- "Item_Visibility"
summary(fdata$Item_Visibility)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## 0.003575 0.031145 0.057194 0.069710 0.096930 0.328391
```

Observation No values with value zero in Item_Visibility variable

NOTE We hypothesized that products with higher visibility are likely to sell more. But along with comparing products on absolute terms, we should look at the visibility of the product in that particular store as compared to the mean visibility of that product across all stores. This will give some idea about how much importance was given to that product in a store as compared to other stores.

```
colnames(fdata)

##  [1] "Item_Identifier"           "Item_Weight"
##  [3] "Item_Fat_Content"          "Item_Visibility"
##  [5] "Item_Type"                 "Item_MRP"
##  [7] "Outlet_Identifier"         "Outlet_Establishment_Year"
##  [9] "Outlet_Size"               "Outlet_Location_Type"
## [11] "Outlet_Type"               "Item_Outlet_Sales"

rm(cdata, i)
cdata <- merge(fdata, avg_Item_Visibility, by="Item_Identifier")
ncol(fdata)

## [1] 12

fdata <- cdata


names(fdata)[names(fdata)=="Item_Visibility.y"] <- "Item_Visibility_MeanRatio
"
names(fdata)[names(fdata)=="Item_Visibility.x"] <- "Item_Visibility"
colnames(fdata)

##  [1] "Item_Identifier"           "Item_Weight"
##  [3] "Item_Fat_Content"          "Item_Visibility"
##  [5] "Item_Type"                 "Item_MRP"
##  [7] "Outlet_Identifier"         "Outlet_Establishment_Year"
##  [9] "Outlet_Size"               "Outlet_Location_Type"
## [11] "Outlet_Type"               "Item_Outlet_Sales"
## [13] "Item_Visibility_MeanRatio"

rm(cdata)
fdata$Item_Visibility_MeanRatio <- as.numeric(fdata$Item_Visibility_MeanRatio
)
class(fdata$Item_Visibility_MeanRatio)

## [1] "numeric"

class(fdata$Item_Visibility)

## [1] "numeric"
```

```r
fdata$Item_Visibility_MeanRatio1 <- fdata$Item_Visibility/fdata$Item_Visibili
ty_MeanRatio
quantile(fdata$Item_Visibility_MeanRatio1)
```

```
##        0%       25%       50%       75%      100% 
## 0.8445628 0.9251308 0.9990698 1.0420067 3.0100939
```

```r
fdata$Item_Visibility_MeanRatio <- fdata$Item_Visibility_MeanRatio1
quantile(fdata$Item_Visibility_MeanRatio1)
```

```
##        0%       25%       50%       75%      100% 
## 0.8445628 0.9251308 0.9990698 1.0420067 3.0100939
```

```r
ncol(fdata)
```

```
## [1] 14
```

```r
fdata <- fdata[, 1:(ncol(fdata)-1)]
head(fdata)
```

```
##    Item_Identifier Item_Weight Item_Fat_Content Item_Visibility
## 1:           DRA12        11.6          Low Fat      0.04094590
## 2:           DRA12        11.6          Low Fat      0.04074762
## 3:           DRA12        11.6               LF      0.04100956
## 4:           DRA12        11.6          Low Fat      0.04117751
## 5:           DRA12        11.6          Low Fat      0.03493779
## 6:           DRA12        11.6          Low Fat      0.04091182
##      Item_Type Item_MRP Outlet_Identifier Outlet_Establishment_Year
## 1: Soft Drinks 142.9154            OUT046                      1997
## 2: Soft Drinks 140.0154            OUT027                      1985
## 3: Soft Drinks 141.0154            OUT049                      1999
## 4: Soft Drinks 140.3154            OUT017                      2007
## 5: Soft Drinks 141.6154            OUT045                      2002
## 6: Soft Drinks 142.3154            OUT013                      1987
##    Outlet_Size Outlet_Location_Type       Outlet_Type Item_Outlet_Sales
## 1:       Small               Tier 1 Supermarket Type1                NA
## 2:      Medium               Tier 3 Supermarket Type3                NA
## 3:      Medium               Tier 1 Supermarket Type1                NA
## 4:       Small               Tier 2 Supermarket Type1          2552.677
## 5:       Small               Tier 2 Supermarket Type1          3829.016
## 6:        High               Tier 3 Supermarket Type1          2552.677
##    Item_Visibility_MeanRatio
## 1:                  1.171966
## 2:                  1.166291
## 3:                  1.173788
## 4:                  1.178595
## 5:                  1.000000
## 6:                  1.170991
```

6.3. Broad category of Type of Item

Earlier we saw that the Item_Type variable has 16 categories which might prove to be very useful in analysis. So its a good idea to combine them. One way could be to manually assign a new category to each. But there???s a catch here. If you look at the Item_Identifier, i.e. the unique ID of each item, it starts with either **F, D or N**. If you see the categories, these look like being Food, Drinks and Non-Consumables. So I???ve used the Item_Identifier variable to create a new column:

```
fdata$Item_Type_Combined <- "NA"

fdata$Item_Type_Combined[grepl("^[fF].*", fdata$Item_Identifier) ] <- "Food"
fdata$Item_Type_Combined[grepl("^[dD].*", fdata$Item_Identifier) ] <- "Drinks
"
fdata$Item_Type_Combined[grepl("^[nN].*", fdata$Item_Identifier) ] <- "Non-Co
nsumable"

table(fdata$Item_Type_Combined)

##
##         Drinks           Food Non-Consumable
##           1317          10201           2686
```

6.4. Determine the years of operation of a store

We wanted to make a new column depicting the years of operation of a store. [NOTE: We are using 2013 Sales Data]

```
fdata$Outlet_Years <- 2013 - fdata$Outlet_Establishment_Year
summary(fdata$Outlet_Years)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4.00    9.00   14.00   15.17   26.00   28.00

table(fdata$Outlet_Years)

##
##    4    6    9   11   14   15   16   26   28
## 1546 1543 1550 1548 1550  925 1550 1553 2439
```

Observation: All the stores are 4-28 years old

6.5. Modify categories of Item_Fat_Content

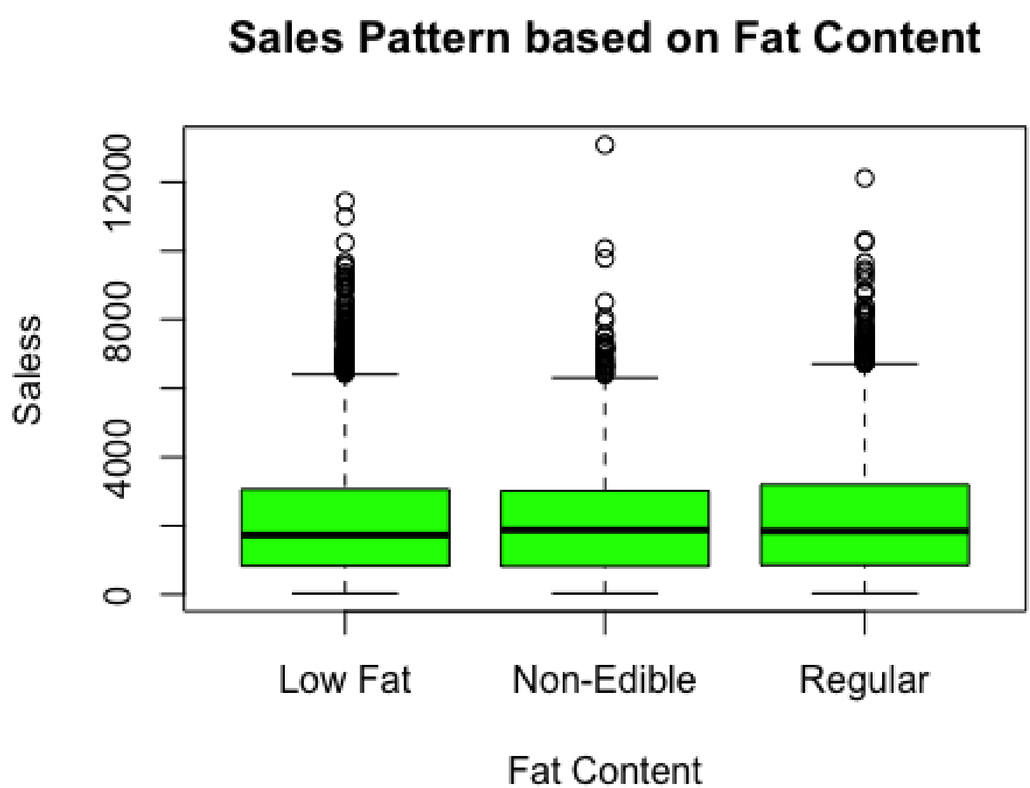
We found typos and difference in representation in categories of Item_Fat_Content variable.

```
fdata$Item_Fat_Content.y <- "NA"
fdata$Item_Fat_Content.y[grepl("^[lL].*", fdata$Item_Fat_Content) ] <- "Low F
at"
fdata$Item_Fat_Content.y[grepl("^[rR].*", fdata$Item_Fat_Content) ] <- "Regul
ar"
```

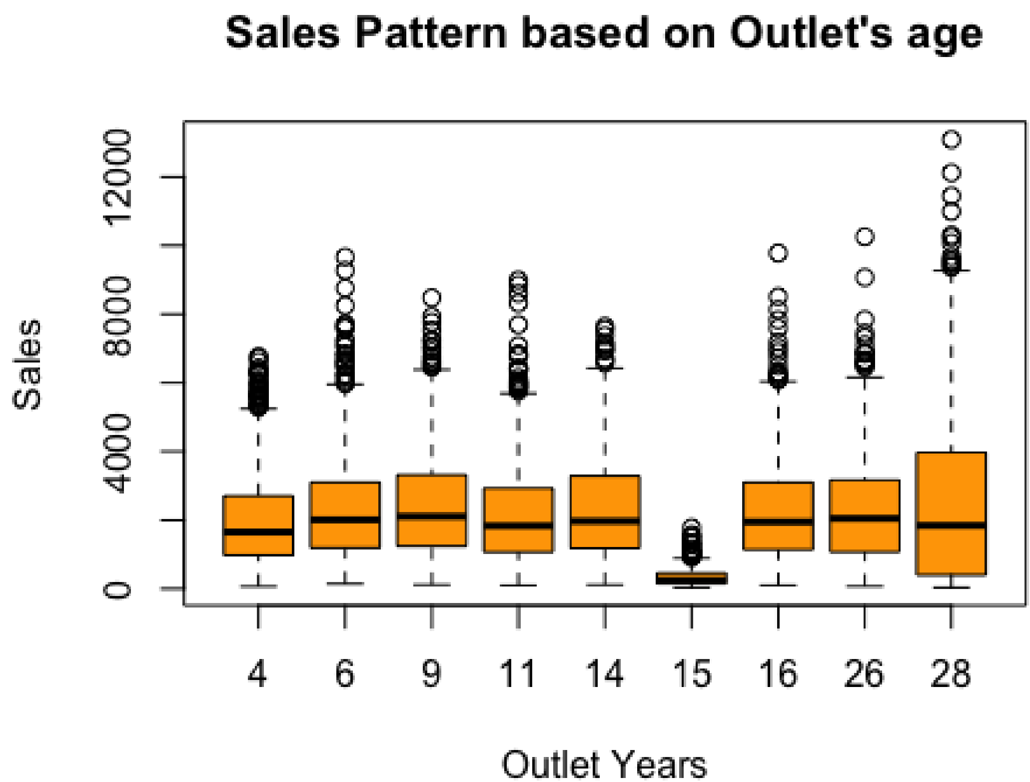
```
fdata$Item_Fat_Content.y[fdata$Item_Type_Combined=="Non-Consumable"] <- "Non-
Edible"
fdata$Item_Fat_Content <- fdata$Item_Fat_Content.y
table(fdata$Item_Fat_Content)

##
##    Low Fat Non-Edible    Regular
##       6499       2686       5019

fdata <- fdata[ ,1:(ncol(fdata)-1)]
#View(fdata)
```

6.6. Exploratory Data Analysis

```
boxplot(fdata$Item_Outlet_Sales~fdata$Item_Fat_Content, xlab="Fat Content", y
lab="Saless", main="Sales Pattern based on Fat Content", col = "green")
```

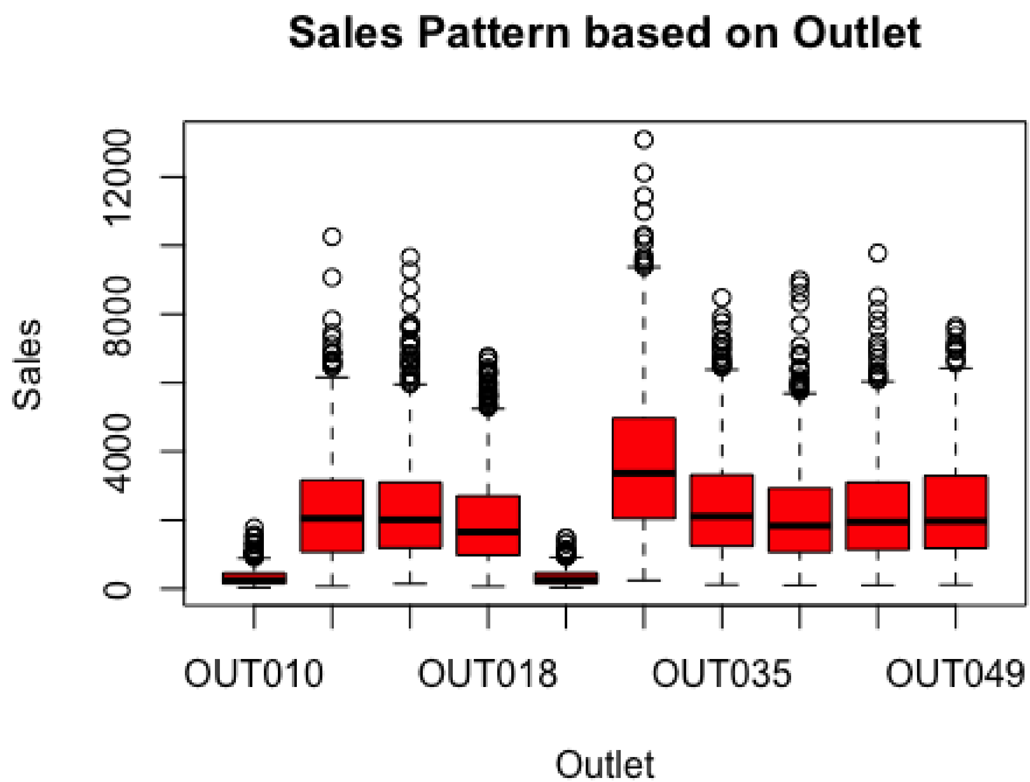



```
boxplot(fdata$Item_Outlet_Sales~fdata$Outlet_Years, xlab="Outlet Years", ylab
="Sales", main="Sales Pattern based on Outlet's age", col = "orange")
```

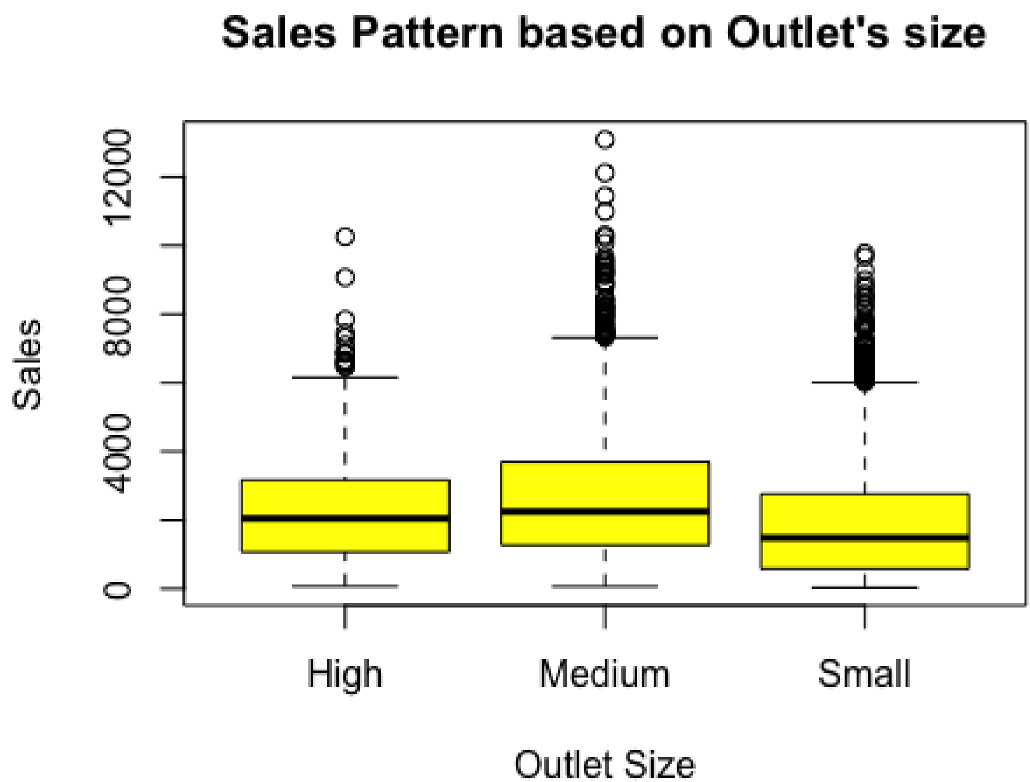



```
boxplot(fdata$Item_Outlet_Sales~fdata$Item_Type_Combined, xlab="Type of Item"
, ylab="Sales", main="Sales Pattern based on type of the item", col = "blue")
```

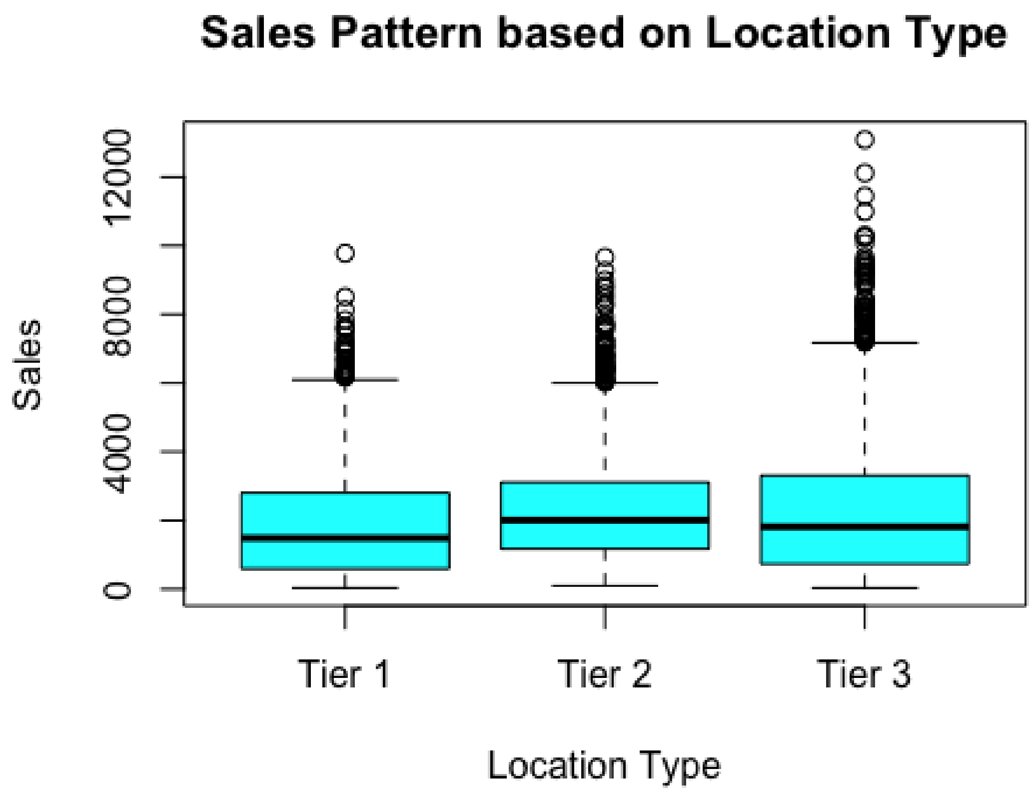

```
boxplot(fdata$Item_Outlet_Sales~fdata$Outlet_Identifier, xlab="Outlet", ylab=
"Sales", main="Sales Pattern based on Outlet", col = "red")
```




```
boxplot(fdata$Item_Outlet_Sales~fdata$Outlet_Size, xlab="Outlet Size", ylab="
Sales", main="Sales Pattern based on Outlet's size", col = "yellow")
```




```
boxplot(fdata$Item_Outlet_Sales~fdata$Outlet_Location_Type, xlab="Location Ty
pe", ylab="Sales", main="Sales Pattern based on Location Type", col = "cyan")
```



Observation: 1. Sales Pattern based on Fat Content: All three performs almost similar 2. Sales Pattern based on Outlet's age: Outlets which are 28 years old performs far better and the outlet which is 16 years old is amongst the worst performers. 3. Sales Pattern based on Type of Item: All three performs almost similar 4. Sales Pattern based on Outlet: Outlet027 outperforms other outlets 5. Sales Pattern based on Outlet's size: The medium sized outlets perform better. 6. Sales Pattern based on Location Type: Tier-3 Performs better as Hypothesized.

6.6. One-Hot Encoding

One-Hot-Coding refers to creating dummy variables, one for each category of a categorical variable.

- For example, the **Item_Fat_Content** has 3 categories ??? ???Low Fat???, ???Regular??? and ???Non-Edible???. One hot coding will remove this variable and generate 3 new variables. Each will have binary numbers ??? 0 (if the category is not present) and 1(if category is present). [Creates **dummy variables**]

- 'Item_Fat_Content'
- 'Outlet_Location_Type'
- 'Outlet_Size'
- 'Item_Type_Combined'

- 'Outlet_Type'
- 'Outlet_Identifier'

NOTE: all columns - Item_Identifier, Item_Weight, Item_Fat_Content, Item_Visibility, Item_Type, Item_MRP, Outlet_Identifier, Outlet_Establishment_Year, Outlet_Size, Outlet_Location_Type, Outlet_Type, Item_Outlet_Sales, Item_Visibility_MeanRatio, Item_Type_Combined, Outlet_Years

```
rm(cdata)
tail(fdata)

##    Item_Identifier Item_Weight Item_Fat_Content Item_Visibility Item_Type
## 1:           NCY18       7.285       Non-Edible      0.03132784 Household
## 2:           NCY18       7.285       Non-Edible      0.05214145 Household
## 3:           NCY18       7.285       Non-Edible      0.03115163 Household
## 4:           NCY18       7.285       Non-Edible      0.03100078 Household
## 5:           NCY18       7.285       Non-Edible      0.03120006 Household
## 6:           NCY18       7.285       Non-Edible      0.03127853 Household
##    Item_MRP Outlet_Identifier Outlet_Establishment_Year Outlet_Size
## 1: 174.6054            OUT017                      2007       Small
## 2: 174.9054            OUT010                      1998       Small
## 3: 173.2054            OUT046                      1997       Small
## 4: 177.0054            OUT027                      1985      Medium
## 5: 174.7054            OUT049                      1999      Medium
## 6: 176.0054            OUT018                      2009      Medium
##    Outlet_Location_Type       Outlet_Type Item_Outlet_Sales
## 1:               Tier 2 Supermarket Type1         2976.7918
## 2:               Tier 3     Grocery Store          525.3162
## 3:               Tier 1 Supermarket Type1         4902.9512
## 4:               Tier 3 Supermarket Type3         2101.2648
## 5:               Tier 1 Supermarket Type1         6303.7944
## 6:               Tier 3 Supermarket Type2         2626.5810
##    Item_Visibility_MeanRatio Item_Type_Combined Outlet_Years
## 1:                 0.9348910     Non-Consumable            6
## 2:                 1.5560144     Non-Consumable           15
## 3:                 0.9296326     Non-Consumable           16
## 4:                 0.9251308     Non-Consumable           28
## 5:                 0.9310779     Non-Consumable           14
## 6:                 0.9334195     Non-Consumable            4

OHECdata <- fdata
#View(OHECdata)

OHECdata <- as.data.frame(OHECdata)
sapply(fdata, function(x) length(unique(x))) #Number of Unique Values in each column

##           Item_Identifier               Item_Weight
##                      1559                       415
##          Item_Fat_Content           Item_Visibility
```

```
##                            3                          13688
##                    Item_Type                       Item_MRP
##                           16                           8052
##            Outlet_Identifier Outlet_Establishment_Year
##                           10                              9
##                  Outlet_Size      Outlet_Location_Type
##                            3                              3
##                  Outlet_Type         Item_Outlet_Sales
##                            4                           3494
## Item_Visibility_MeanRatio        Item_Type_Combined
##                        13287                              3
##                 Outlet_Years
##                            9
```

```
#write.csv(fdata, "final_data.csv")

#Item_Fat_Content
OHECdata <- with(OHECdata,
       data.frame(Item_Identifier, Item_Weight, Item_Visibility, Item_Type, I
tem_Fat_Content, Item_MRP, Outlet_Identifier, Outlet_Establishment_Year, Outl
et_Size, Outlet_Location_Type, Outlet_Type, Item_Outlet_Sales, Item_Visibilit
y_MeanRatio, Item_Type_Combined, Outlet_Years, model.matrix(~Item_Fat_Content
-1,OHECdata)))

head(OHECdata)
```

```
##   Item_Identifier Item_Weight Item_Visibility   Item_Type Item_Fat_Content
## 1           DRA12        11.6      0.04094590 Soft Drinks          Low Fat
## 2           DRA12        11.6      0.04074762 Soft Drinks          Low Fat
## 3           DRA12        11.6      0.04100956 Soft Drinks          Low Fat
## 4           DRA12        11.6      0.04117751 Soft Drinks          Low Fat
## 5           DRA12        11.6      0.03493779 Soft Drinks          Low Fat
## 6           DRA12        11.6      0.04091182 Soft Drinks          Low Fat
##   Item_MRP Outlet_Identifier Outlet_Establishment_Year Outlet_Size
## 1 142.9154            OUT046                      1997       Small
## 2 140.0154            OUT027                      1985      Medium
## 3 141.0154            OUT049                      1999      Medium
## 4 140.3154            OUT017                      2007       Small
## 5 141.6154            OUT045                      2002       Small
## 6 142.3154            OUT013                      1987        High
##   Outlet_Location_Type       Outlet_Type Item_Outlet_Sales
## 1               Tier 1 Supermarket Type1                NA
## 2               Tier 3 Supermarket Type3                NA
## 3               Tier 1 Supermarket Type1                NA
## 4               Tier 2 Supermarket Type1          2552.677
## 5               Tier 2 Supermarket Type1          3829.016
## 6               Tier 3 Supermarket Type1          2552.677
##   Item_Visibility_MeanRatio Item_Type_Combined Outlet_Years
## 1                  1.171966             Drinks           16
## 2                  1.166291             Drinks           28
```

```
## 3                 1.173788             Drinks          14
## 4                 1.178595             Drinks           6
## 5                 1.000000             Drinks          11
## 6                 1.170991             Drinks          26
##   Item_Fat_ContentLow.Fat Item_Fat_ContentNon.Edible
## 1                       1                          0
## 2                       1                          0
## 3                       1                          0
## 4                       1                          0
## 5                       1                          0
## 6                       1                          0
##   Item_Fat_ContentRegular
## 1                       0
## 2                       0
## 3                       0
## 4                       0
## 5                       0
## 6                       0
```

```
#View(OHECdata)
```

Observation: New Columns added are as follows:-

1. Item_Fat_ContentLow.Fat
2. Item_Fat_ContentNon.Edible
3. Item_Fat_ContentRegular

```
#Outlet_Location_Type
OHECdata <- with(OHECdata,
       data.frame(Item_Identifier, Item_Weight, Item_Visibility, Item_Type, I
tem_Fat_Content, Outlet_Location_Type, Item_MRP, Outlet_Identifier, Outlet_Es
tablishment_Year, Outlet_Size, Outlet_Type, Item_Outlet_Sales, Item_Visibilit
y_MeanRatio, Item_Type_Combined, Outlet_Years, Item_Fat_ContentLow.Fat, Item_
Fat_ContentNon.Edible, Item_Fat_ContentRegular, model.matrix(~Outlet_Location
_Type-1,OHECdata)))

head(OHECdata)
```

```
##   Item_Identifier Item_Weight Item_Visibility   Item_Type Item_Fat_Content
## 1           DRA12        11.6      0.04094590 Soft Drinks          Low Fat
## 2           DRA12        11.6      0.04074762 Soft Drinks          Low Fat
## 3           DRA12        11.6      0.04100956 Soft Drinks          Low Fat
## 4           DRA12        11.6      0.04117751 Soft Drinks          Low Fat
## 5           DRA12        11.6      0.03493779 Soft Drinks          Low Fat
## 6           DRA12        11.6      0.04091182 Soft Drinks          Low Fat
##   Outlet_Location_Type Item_MRP Outlet_Identifier
## 1               Tier 1 142.9154            OUT046
## 2               Tier 3 140.0154            OUT027
## 3               Tier 1 141.0154            OUT049
## 4               Tier 2 140.3154            OUT017
## 5               Tier 2 141.6154            OUT045
```

```
## 6                Tier 3 142.3154            OUT013
##   Outlet_Establishment_Year Outlet_Size       Outlet_Type
## 1                      1997       Small Supermarket Type1
## 2                      1985      Medium Supermarket Type3
## 3                      1999      Medium Supermarket Type1
## 4                      2007       Small Supermarket Type1
## 5                      2002       Small Supermarket Type1
## 6                      1987        High Supermarket Type1
##   Item_Outlet_Sales Item_Visibility_MeanRatio Item_Type_Combined
## 1                NA                  1.171966             Drinks
## 2                NA                  1.166291             Drinks
## 3                NA                  1.173788             Drinks
## 4          2552.677                  1.178595             Drinks
## 5          3829.016                  1.000000             Drinks
## 6          2552.677                  1.170991             Drinks
##   Outlet_Years Item_Fat_ContentLow.Fat Item_Fat_ContentNon.Edible
## 1           16                       1                          0
## 2           28                       1                          0
## 3           14                       1                          0
## 4            6                       1                          0
## 5           11                       1                          0
## 6           26                       1                          0
##   Item_Fat_ContentRegular Outlet_Location_TypeTier.1
## 1                       0                          1
## 2                       0                          0
## 3                       0                          1
## 4                       0                          0
## 5                       0                          0
## 6                       0                          0
##   Outlet_Location_TypeTier.2 Outlet_Location_TypeTier.3
## 1                          0                          0
## 2                          0                          1
## 3                          0                          0
## 4                          1                          0
## 5                          1                          0
## 6                          0                          1

#View(OHECdata)
```

Observation: New Columns added are as follows:-

1. Outlet_Location_TypeTier.1
2. Outlet_Location_TypeTier.2
3. Outlet_Location_TypeTier.3

```
#Outlet_Size
OHECdata <- with(OHECdata,
      data.frame(Item_Identifier, Item_Weight, Item_Visibility, Item_Type, I
tem_Fat_Content, Outlet_Location_Type, Outlet_Size, Item_MRP, Outlet_Identifi
er, Outlet_Establishment_Year, Outlet_Type, Item_Outlet_Sales, Item_Visibilit
y_MeanRatio, Item_Type_Combined, Outlet_Years, Item_Fat_ContentLow.Fat, Item_
```

```
Fat_ContentNon.Edible, Item_Fat_ContentRegular, Outlet_Location_TypeTier.1, O
utlet_Location_TypeTier.2, Outlet_Location_TypeTier.3, model.matrix(~Outlet_S
ize-1,OHECdata)))

#head(OHECdata)
#View(OHECdata)
```

Observation: New Columns added are as follows:-

1. Outlet_SizeHigh
2. Outlet_SizeMedium
3. Outlet_SizeSmall

```
#Item_Type_Combined
OHECdata <- with(OHECdata,
       data.frame(Item_Identifier, Item_Weight, Item_Visibility, Item_Type, I
tem_Fat_Content, Outlet_Location_Type, Outlet_Size, Item_Type_Combined, Item_
MRP, Outlet_Identifier, Outlet_Establishment_Year, Outlet_Type, Item_Outlet_S
ales, Item_Visibility_MeanRatio, Outlet_Years, Item_Fat_ContentLow.Fat, Item_
Fat_ContentNon.Edible, Item_Fat_ContentRegular, Outlet_Location_TypeTier.1, O
utlet_Location_TypeTier.2, Outlet_Location_TypeTier.3, Outlet_SizeHigh, Outle
t_SizeMedium, Outlet_SizeSmall, model.matrix(~Item_Type_Combined-1,OHECdata))
)

#head(OHECdata)
#View(OHECdata)
```

Observation: New Columns added are as follows:-

1. Item_Type_CombinedDrinks
2. Item_Type_CombinedFood
3. Item_Type_CombinedNon.Consumable

```
#Outlet_Type
OHECdata <- with(OHECdata,
       data.frame(Item_Identifier, Item_Weight, Item_Visibility, Item_Type, I
tem_Fat_Content, Outlet_Location_Type, Outlet_Size, Item_Type_Combined, Outle
t_Type, Item_MRP, Outlet_Identifier, Outlet_Establishment_Year, Item_Outlet_S
ales, Item_Visibility_MeanRatio, Outlet_Years, Item_Fat_ContentLow.Fat, Item_
Fat_ContentNon.Edible, Item_Fat_ContentRegular, Outlet_Location_TypeTier.1, O
utlet_Location_TypeTier.2, Outlet_Location_TypeTier.3, Outlet_SizeHigh, Outle
t_SizeMedium, Outlet_SizeSmall, Item_Type_CombinedDrinks, Item_Type_CombinedF
ood, Item_Type_CombinedNon.Consumable, model.matrix(~Outlet_Type-1,OHECdata))
)

#head(OHECdata)
#View(OHECdata)
```

Observation: New Columns added are as follows:-

1. Outlet_TypeGrocery.Store

2. Outlet_TypeSupermarket.Type1
3. Outlet_TypeSupermarket.Type2
4. Outlet_TypeSupermarket.Type3

```
#Outlet_Identifier

final_data <- OHECdata

OHECdata <- with(OHECdata,
       data.frame(Item_Identifier, Item_Weight, Item_Visibility, Item_Type, I
tem_Fat_Content, Outlet_Location_Type, Outlet_Size, Item_Type_Combined, Outle
t_Type, Item_MRP, Outlet_Identifier, Outlet_Establishment_Year, Item_Outlet_S
ales, Item_Visibility_MeanRatio, Outlet_Years, Item_Fat_ContentLow.Fat, Item_
Fat_ContentNon.Edible, Item_Fat_ContentRegular, Outlet_Location_TypeTier.1, O
utlet_Location_TypeTier.2, Outlet_Location_TypeTier.3, Outlet_SizeHigh, Outle
t_SizeMedium, Outlet_SizeSmall, Item_Type_CombinedDrinks, Item_Type_CombinedF
ood, Item_Type_CombinedNon.Consumable, Outlet_TypeGrocery.Store, Outlet_TypeS
upermarket.Type1, Outlet_TypeSupermarket.Type2, Outlet_TypeSupermarket.Type3,
model.matrix(~Outlet_Identifier-1,OHECdata)))

#head(OHECdata)
#View(OHECdata)
```

Observation: Nine columns are added, each indicating the unique outlet identifier. With this, we can find which outlet has made most of the sales.

6.6.1. One-Hot Encoding - Validate

Lets look at the 3 columns formed from Item_Fat_Content

```
OHECdata <- as.data.table(OHECdata)

head(cbind(OHECdata$Item_Fat_ContentLow.Fat, OHECdata$Item_Fat_ContentNon.Edi
ble, OHECdata$Item_Fat_ContentRegular), 20)

##       [,1] [,2] [,3]
##  [1,]    1    0    0
##  [2,]    1    0    0
##  [3,]    1    0    0
##  [4,]    1    0    0
##  [5,]    1    0    0
##  [6,]    1    0    0
##  [7,]    1    0    0
##  [8,]    1    0    0
##  [9,]    1    0    0
## [10,]    0    0    1
## [11,]    0    0    1
## [12,]    0    0    1
## [13,]    0    0    1
## [14,]    0    0    1
## [15,]    0    0    1
```

```
## [16,]    0    0    1
## [17,]    0    0    1
## [18,]    0    0    1
## [19,]    0    0    1
## [20,]    0    0    1
```

Observation: We can see the binary values in the columns - One Hot Encoding worked!

7. Exporting Data

Let us now export the dataset as follows:

1. Remove the unnecessary columns - Item_Type, Establishment_Year
2. Partition the data-set in such a way that the test data-set should not have the target variable or the dependent variable
3. All other independent variables to be present in both the test data set and the train data set.
4. In addition to the independent variables, the train data-set should also have the target variable or the dependent variable.

```
OHECdata <- as.data.frame(OHECdata)
drop_columns <- c("Item_Type","Outlet_Establishment_Year")
Export_data <- OHECdata[ , !(names(OHECdata) %in% drop_columns)]

head(Export_data)

##   Item_Identifier Item_Weight Item_Visibility Item_Fat_Content
## 1           DRA12        11.6      0.04094590          Low Fat
## 2           DRA12        11.6      0.04074762          Low Fat
## 3           DRA12        11.6      0.04100956          Low Fat
## 4           DRA12        11.6      0.04117751          Low Fat
## 5           DRA12        11.6      0.03493779          Low Fat
## 6           DRA12        11.6      0.04091182          Low Fat
##   Outlet_Location_Type Outlet_Size Item_Type_Combined       Outlet_Type
## 1               Tier 1       Small             Drinks Supermarket Type1
## 2               Tier 3      Medium             Drinks Supermarket Type3
## 3               Tier 1      Medium             Drinks Supermarket Type1
## 4               Tier 2       Small             Drinks Supermarket Type1
## 5               Tier 2       Small             Drinks Supermarket Type1
## 6               Tier 3        High             Drinks Supermarket Type1
##   Item_MRP Outlet_Identifier Item_Outlet_Sales Item_Visibility_MeanRatio
## 1 142.9154            OUT046                NA                  1.171966
## 2 140.0154            OUT027                NA                  1.166291
## 3 141.0154            OUT049                NA                  1.173788
## 4 140.3154            OUT017          2552.677                  1.178595
## 5 141.6154            OUT045          3829.016                  1.000000
## 6 142.3154            OUT013          2552.677                  1.170991
##   Outlet_Years Item_Fat_ContentLow.Fat Item_Fat_ContentNon.Edible
## 1           16                       1                          0
## 2           28                       1                          0
```

```
## 3          14                   1                    0
## 4           6                   1                    0
## 5          11                   1                    0
## 6          26                   1                    0
##   Item_Fat_ContentRegular Outlet_Location_TypeTier.1
## 1                       0                          1
## 2                       0                          0
## 3                       0                          1
## 4                       0                          0
## 5                       0                          0
## 6                       0                          0
##   Outlet_Location_TypeTier.2 Outlet_Location_TypeTier.3 Outlet_SizeHigh
## 1                          0                          0               0
## 2                          0                          1               0
## 3                          0                          0               0
## 4                          1                          0               0
## 5                          1                          0               0
## 6                          0                          1               1
##   Outlet_SizeMedium Outlet_SizeSmall Item_Type_CombinedDrinks
## 1                 0                1                        1
## 2                 1                0                        1
## 3                 1                0                        1
## 4                 0                1                        1
## 5                 0                1                        1
## 6                 0                0                        1
##   Item_Type_CombinedFood Item_Type_CombinedNon.Consumable
## 1                      0                                0
## 2                      0                                0
## 3                      0                                0
## 4                      0                                0
## 5                      0                                0
## 6                      0                                0
##   Outlet_TypeGrocery.Store Outlet_TypeSupermarket.Type1
## 1                        0                            1
## 2                        0                            0
## 3                        0                            1
## 4                        0                            1
## 5                        0                            1
## 6                        0                            1
##   Outlet_TypeSupermarket.Type2 Outlet_TypeSupermarket.Type3
## 1                            0                            0
## 2                            0                            1
## 3                            0                            0
## 4                            0                            0
## 5                            0                            0
## 6                            0                            0
##   Outlet_IdentifierOUT010 Outlet_IdentifierOUT013 Outlet_IdentifierOUT017
## 1                       0                       0                       0
## 2                       0                       0                       0
## 3                       0                       0                       0
```

```
## 4                        0                        0                        1
## 5                        0                        0                        0
## 6                        0                        1                        0
##   Outlet_IdentifierOUT018 Outlet_IdentifierOUT019 Outlet_IdentifierOUT027
## 1                        0                        0                        0
## 2                        0                        0                        1
## 3                        0                        0                        0
## 4                        0                        0                        0
## 5                        0                        0                        0
## 6                        0                        0                        0
##   Outlet_IdentifierOUT035 Outlet_IdentifierOUT045 Outlet_IdentifierOUT046
## 1                        0                        0                        1
## 2                        0                        0                        0
## 3                        0                        0                        0
## 4                        0                        0                        0
## 5                        0                        1                        0
## 6                        0                        0                        0
##   Outlet_IdentifierOUT049
## 1                        0
## 2                        0
## 3                        1
## 4                        0
## 5                        0
## 6                        0

Export_data <- as.data.table(Export_data)

test_Export <- Export_data[is.na(Item_Outlet_Sales), ]
train_Export <- Export_data[!is.na(Item_Outlet_Sales), ]

# write.csv(Export_data, "data_Export.csv")
# write.csv(test_Export, "test_Export.csv")
# write.csv(train_Export, "train_Export.csv")
rm(list = ls())
```

8. Reading data

Now let us read the train and the test dataset separately for the purpose of model building.

```
rm(list=ls())
train <- read.csv("train_Export.csv", header=T, na.strings=c("","NA"))
test <- read.csv("test_Export.csv", header=T, na.strings=c("","NA"))
fdata <- read.csv("data_Export.csv", header=T, na.strings=c("","NA"))

train <- as.data.table(train)
test <- as.data.table(test)
fdata <- as.data.table(fdata)

glimpse(train)
```

```
## Observations: 8,523
## Variables: 40
## $ X                              <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10...
## $ Item_Identifier                <fctr> DRA12, DRA12, DRA12, DRA12, ...
## $ Item_Weight                    <dbl> 11.600, 11.600, 11.600, 11.60...
## $ Item_Visibility                <dbl> 0.041177505, 0.034937793, 0.0...
## $ Item_Fat_Content               <fctr> Low Fat, Low Fat, Low Fat, L...
## $ Outlet_Location_Type           <fctr> Tier 2, Tier 2, Tier 3, Tier...
## $ Outlet_Size                    <fctr> Small, Small, High, Small, M...
## $ Item_Type_Combined             <fctr> Drinks, Drinks, Drinks, Drin...
## $ Outlet_Type                    <fctr> Supermarket Type1, Supermark...
## $ Item_MRP                       <dbl> 140.3154, 141.6154, 142.3154,...
## $ Outlet_Identifier              <fctr> OUT017, OUT045, OUT013, OUT0...
## $ Item_Outlet_Sales              <dbl> 2552.6772, 3829.0158, 2552.67...
## $ Item_Visibility_MeanRatio      <dbl> 1.1785949, 1.0000000, 1.17099...
## $ Outlet_Years                   <int> 6, 11, 26, 9, 4, 15, 6, 28, 1...
## $ Item_Fat_ContentLow.Fat        <int> 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,...
## $ Item_Fat_ContentNon.Edible     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Item_Fat_ContentRegular        <int> 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,...
## $ Outlet_Location_TypeTier.1     <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Outlet_Location_TypeTier.2     <int> 1, 1, 0, 1, 0, 0, 1, 0, 0, 0,...
## $ Outlet_Location_TypeTier.3     <int> 0, 0, 1, 0, 1, 1, 0, 0, 1, 1,...
## $ Outlet_SizeHigh                <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,...
## $ Outlet_SizeMedium              <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,...
## $ Outlet_SizeSmall               <int> 1, 1, 0, 1, 0, 1, 1, 1, 1, 0,...
## $ Item_Type_CombinedDrinks       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Item_Type_CombinedFood         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Item_Type_CombinedNon.Consumable <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Outlet_TypeGrocery.Store       <int> 0, 0, 0, 0, 0, 1, 0, 1, 1, 0,...
## $ Outlet_TypeSupermarket.Type1   <int> 1, 1, 1, 1, 0, 0, 1, 0, 0, 0,...
## $ Outlet_TypeSupermarket.Type2   <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,...
## $ Outlet_TypeSupermarket.Type3   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,...
## $ Outlet_IdentifierOUT010        <int> 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,...
## $ Outlet_IdentifierOUT013        <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,...
## $ Outlet_IdentifierOUT017        <int> 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,...
## $ Outlet_IdentifierOUT018        <int> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,...
## $ Outlet_IdentifierOUT019        <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,...
## $ Outlet_IdentifierOUT027        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,...
## $ Outlet_IdentifierOUT035        <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,...
## $ Outlet_IdentifierOUT045        <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Outlet_IdentifierOUT046        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ Outlet_IdentifierOUT049        <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
```

We can see that the data is properly exported on performing one-hot-encoding (with 0's and 1's indicating its presence). Now that we have the data ready, its time to start making predictive models.

8.1. Baseline Model

Baseline model is the one which requires no predictive model and its like an informed guess. For instance, in this case lets predict the sales as the overall average sales.

NOTE: If the score of the predictive algorithm is below this, then there is something going seriously wrong and the data is to be checked.

```
#Mean based:

mean_sales <- mean(train$Item_Outlet_Sales)

drop_columns <- c("X", "Item_Identifier", "Outlet_Identifier", "Item_Outlet_S
ales")
baseline_model <- test[,!(names(test) %in% drop_columns)] #input_variables_va
lues_training_datasets
baseline_model$Item_Outlet_Sales <- mean_sales
```

Observation: We can see that every observation in the **Item_Outlet_Sales** is predicted to be 2181.29. This is the average or mean of the Item_Outlet_Sales. Thus, gives a very poor model. The aim of this model is to have a benchamark below which our subsequent models should not perform.

8.2. Decision Trees

```
train <- as.data.frame(train)
library(rpart)     # Decision Trees

dt <- rpart(Item_Outlet_Sales ~ Outlet_IdentifierOUT046 + Outlet_IdentifierOU
T045 + Outlet_IdentifierOUT049
                   + Outlet_IdentifierOUT035 + Outlet_IdentifierOUT018 + Outl
et_IdentifierOUT019 + Outlet_IdentifierOUT027
                   + Outlet_IdentifierOUT017 + Outlet_IdentifierOUT013 + Outl
et_IdentifierOUT010 + Outlet_TypeSupermarket.Type3
                   + Outlet_TypeSupermarket.Type2 + Outlet_TypeSupermarket.Ty
pe1 + Item_Type_CombinedFood +
                     Item_Type_CombinedNon.Consumable + Outlet_TypeGrocery.St
ore + Item_Type_CombinedDrinks + Outlet_SizeSmall
                   + Outlet_SizeMedium + Outlet_Location_TypeTier.2 + Outlet_
Location_TypeTier.3 + Outlet_Location_TypeTier.1
                   + Outlet_SizeHigh + Item_Fat_ContentRegular + Item_Fat_Con
tentNon.Edible + Item_Fat_ContentLow.Fat
                   + Outlet_Years + Item_Visibility_MeanRatio + Item_MRP, dat
a = train, method = "anova")

plot(dt)
text(dt, pretty = 0, cex = 0.5)
summary(dt)
drop_columns <- c("X", "Item_Identifier", "Outlet_Identifier", "Item_Outlet_S
ales")
dt_test <- test[,!(names(test) %in% drop_columns)] #input_variables_values_tr
```

```
aining_datasets
class(dt)

predicted_sales_dt <- predict(dt, dt_test)
head(predicted_sales_dt)
#dt_test$Item_Outlet_Sales <- predicted_sales_dt
```

Observation:

Variable importance: (most important variable at 1.)

1. **Item_MRP:** Price of the item

2. **Outlet_TypeGrocery.Store:** Outlet type is Grocery Store

3. **Item_Visibility_MeanRatio:** Space given for the item at the display

4. **Outlet_IdentifierOUT010:** Unique outlet identifier (there are 9 outlets involved in this analysis)
5. **Outlet_IdentifierOUT019:** Unique outlet identifier (there are 9 outlets involved in this analysis)

6. **Outlet_Years:** Number of years since the outlet is opened

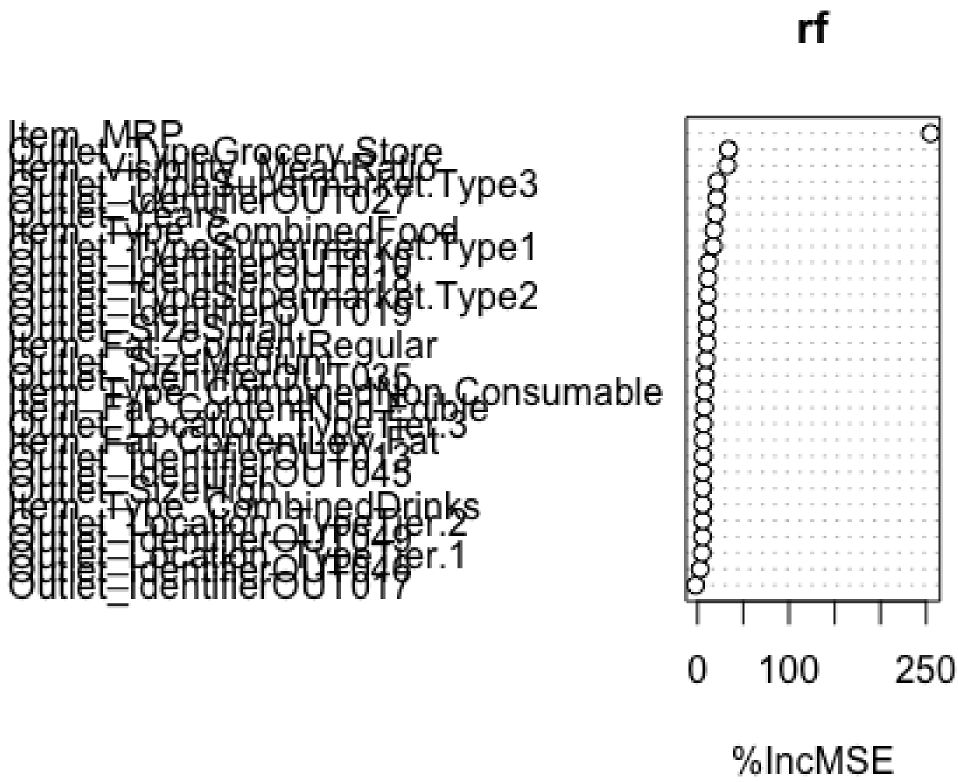
7. **Outlet_IdentifierOUT027:** Unique outlet identifier (there are 9 outlets involved in this analysis)
8. **Outlet_TypeSupermarket.Type3:** Outlet Type is Super-Market Type 3

Further, we have predicted the Item_Outlet_Sales based on this decision tree model and have stored. The rmse and cp for the decision tree is computed and displayed at the end (along with the model comparison chunk)

8.3. Random Forest

```
library(randomForest)
train <- as.data.frame(train)

rf <- randomForest(Item_Outlet_Sales ~ Outlet_IdentifierOUT046 + Outlet_Ident
ifierOUT045 + Outlet_IdentifierOUT049
                   + Outlet_IdentifierOUT035 + Outlet_IdentifierOUT018 + Outl
et_IdentifierOUT019 + Outlet_IdentifierOUT027
                   + Outlet_IdentifierOUT017 + Outlet_IdentifierOUT013 + Outl
et_IdentifierOUT010 + Outlet_TypeSupermarket.Type3
                   + Outlet_TypeSupermarket.Type2 + Outlet_TypeSupermarket.Ty
pe1 + Item_Type_CombinedFood +
                     Item_Type_CombinedNon.Consumable + Outlet_TypeGrocery.St
ore + Item_Type_CombinedDrinks + Outlet_SizeSmall
                   + Outlet_SizeMedium + Outlet_Location_TypeTier.2 + Outlet_
Location_TypeTier.3 + Outlet_Location_TypeTier.1
```

```
                   + Outlet_SizeHigh + Item_Fat_ContentRegular + Item_Fat_Con
tentNon.Edible + Item_Fat_ContentLow.Fat
                   + Outlet_Years + Item_Visibility_MeanRatio + Item_MRP, dat
a = train, importance = TRUE, ntree=1000)
which.min(rf$mse)

## [1] 990

imp <- as.data.frame(sort(importance(rf)[,1],decreasing = TRUE),optional = T)
names(imp) <- "% Inc MSE"
imp

##                                   % Inc MSE
## Item_MRP                         254.863875
## Outlet_TypeGrocery.Store          33.790556
## Item_Visibility_MeanRatio         32.314414
## Outlet_TypeSupermarket.Type3      21.452405
## Outlet_IdentifierOUT027           20.993654
## Outlet_Years                      19.839371
## Item_Type_CombinedFood            17.763837
## Outlet_TypeSupermarket.Type1      16.980811
## Outlet_IdentifierOUT010           12.069902
## Outlet_IdentifierOUT018           11.415096
## Outlet_TypeSupermarket.Type2      11.150438
## Outlet_IdentifierOUT019           10.510214
## Outlet_SizeSmall                  10.369831
## Item_Fat_ContentRegular           10.251719
## Outlet_SizeMedium                  9.331654
## Outlet_IdentifierOUT035            8.533123
## Item_Type_CombinedNon.Consumable   7.743783
## Item_Fat_ContentNon.Edible         7.386312
## Outlet_Location_TypeTier.3         6.363780
## Item_Fat_ContentLow.Fat            6.178120
## Outlet_IdentifierOUT013            5.965258
## Outlet_IdentifierOUT045            5.779737
## Outlet_SizeHigh                    5.663066
## Item_Type_CombinedDrinks           5.618354
## Outlet_Location_TypeTier.2         5.502928
## Outlet_IdentifierOUT049            5.447064
## Outlet_Location_TypeTier.1         4.961511
## Outlet_IdentifierOUT046            2.788518
## Outlet_IdentifierOUT017           -2.146108

varImpPlot(rf, sort = TRUE, type = 1)
```




```r
test <- as.data.frame(test)

drop_columns <- c("X", "Item_Identifier", "Outlet_Identifier", "Item_Outlet_S
ales")
rf_test <- test[,!(names(test) %in% drop_columns)] #input_variables_values_tr
aining_datasets


predicted_sales_rf <- predict(rf, rf_test)
rf_test$Item_Outlet_Sales <- predicted_sales_rf
```

Observation:

1. It is not suprising to see that the variable importance predicted by decision tree and Random Forest is almost the same. (Random Forest is just the collection of Decision Trees)

- train$Item_MRP 280.203753

- train$Outlet_Type 38.471388

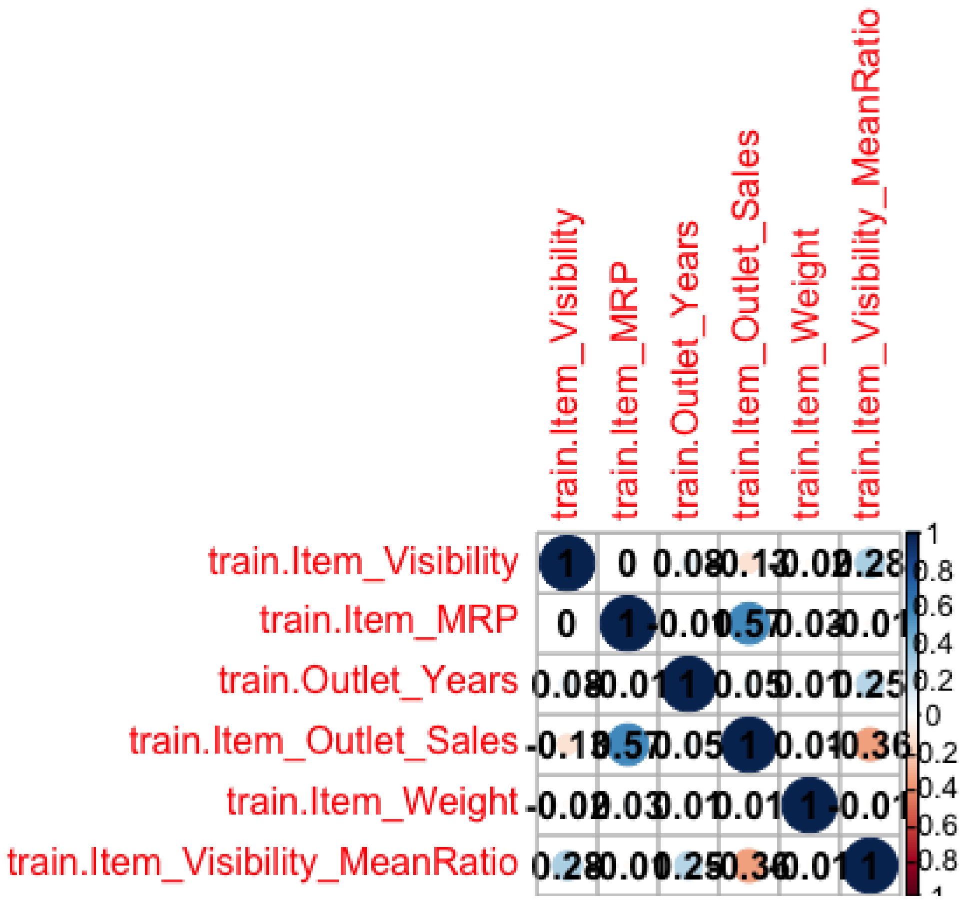
- train$Outlet_Identifier 35.830600

- train$Outlet_Years 28.831678
- train$Outlet_Size 17.156380
- train$Item_Visibility 14.210743
- train$Outlet_Location_Type 10.665934
- train$Item_Weight 5.783006
- train$Item_Fat_Content 3.132697

2.We have predicted the Item_Outlet_Sales based on this Random Forest model and have stored.

8.4. Linear Regression Model

```
library(plyr)
library(dplyr)
library(randomForest)
library(corrplot)
colnames(train)

##  [1] "X"                                "Item_Identifier"
##  [3] "Item_Weight"                      "Item_Visibility"
##  [5] "Item_Fat_Content"                 "Outlet_Location_Type"
##  [7] "Outlet_Size"                      "Item_Type_Combined"
##  [9] "Outlet_Type"                      "Item_MRP"
## [11] "Outlet_Identifier"                "Item_Outlet_Sales"
## [13] "Item_Visibility_MeanRatio"        "Outlet_Years"
## [15] "Item_Fat_ContentLow.Fat"          "Item_Fat_ContentNon.Edible"
## [17] "Item_Fat_ContentRegular"          "Outlet_Location_TypeTier.1"
## [19] "Outlet_Location_TypeTier.2"       "Outlet_Location_TypeTier.3"
## [21] "Outlet_SizeHigh"                  "Outlet_SizeMedium"
## [23] "Outlet_SizeSmall"                 "Item_Type_CombinedDrinks"
## [25] "Item_Type_CombinedFood"           "Item_Type_CombinedNon.Consumable"
## [27] "Outlet_TypeGrocery.Store"         "Outlet_TypeSupermarket.Type1"
## [29] "Outlet_TypeSupermarket.Type2"     "Outlet_TypeSupermarket.Type3"
## [31] "Outlet_IdentifierOUT010"          "Outlet_IdentifierOUT013"
## [33] "Outlet_IdentifierOUT017"          "Outlet_IdentifierOUT018"
## [35] "Outlet_IdentifierOUT019"          "Outlet_IdentifierOUT027"
## [37] "Outlet_IdentifierOUT035"          "Outlet_IdentifierOUT045"
## [39] "Outlet_IdentifierOUT046"          "Outlet_IdentifierOUT049"

sub=data.frame(train$Item_Visibility,train$Item_MRP,train$Outlet_Years, train
$Item_Outlet_Sales, train$Item_Weight, train$Item_Visibility_MeanRatio)
sub <- cor(sub)
corrplot(sub, method="circle", addCoef.col="black")
```



Observation:

1. Based on the correlation plot we can observe that Item_MRP is strongly correlated to the Item_Outlet_Sales: This is in-line with our hypotheses.

2. Further we can see that the Item's Visibility ratio is negavtively correlated with the Item_Outlet_Sales: This is not in-line with our hypotheses.

```
train <- as.data.frame(train)

linear_model <- lm(Item_Outlet_Sales ~ Outlet_IdentifierOUT046 + Outlet_Ident
ifierOUT045 + Outlet_IdentifierOUT049
                   + Outlet_IdentifierOUT035 + Outlet_IdentifierOUT018 + Outl
et_IdentifierOUT019 + Outlet_IdentifierOUT027
                   + Outlet_IdentifierOUT017 + Outlet_IdentifierOUT013 + Outl
et_IdentifierOUT010 + Outlet_TypeSupermarket.Type3
                   + Outlet_TypeSupermarket.Type2 + Outlet_TypeSupermarket.Ty
pe1 + Item_Type_CombinedFood +
                     Item_Type_CombinedNon.Consumable + Outlet_TypeGrocery.St
ore + Item_Type_CombinedDrinks + Outlet_SizeSmall
                   + Outlet_SizeMedium + Outlet_Location_TypeTier.2 + Outlet_
```

```
Location_TypeTier.3 + Outlet_Location_TypeTier.1
                   + Outlet_SizeHigh + Item_Fat_ContentRegular + Item_Fat_Con
tentNon.Edible + Item_Fat_ContentLow.Fat
                   + Outlet_Years + Item_Visibility_MeanRatio + Item_MRP, dat
a = train)
summary(linear_model)
```

```
##
## Call:
## lm(formula = Item_Outlet_Sales ~ Outlet_IdentifierOUT046 + Outlet_Identifi
erOUT045 +
##     Outlet_IdentifierOUT049 + Outlet_IdentifierOUT035 + Outlet_IdentifierO
UT018 +
##     Outlet_IdentifierOUT019 + Outlet_IdentifierOUT027 + Outlet_IdentifierO
UT017 +
##     Outlet_IdentifierOUT013 + Outlet_IdentifierOUT010 + Outlet_TypeSuperma
rket.Type3 +
##     Outlet_TypeSupermarket.Type2 + Outlet_TypeSupermarket.Type1 +
##     Item_Type_CombinedFood + Item_Type_CombinedNon.Consumable +
##     Outlet_TypeGrocery.Store + Item_Type_CombinedDrinks + Outlet_SizeSmall
+
##     Outlet_SizeMedium + Outlet_Location_TypeTier.2 + Outlet_Location_TypeT
ier.3 +
##     Outlet_Location_TypeTier.1 + Outlet_SizeHigh + Item_Fat_ContentRegular
+
##     Item_Fat_ContentNon.Edible + Item_Fat_ContentLow.Fat + Outlet_Years +
##     Item_Visibility_MeanRatio + Item_MRP, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4331.8  -677.8   -88.9   572.5  7942.5
##
## Coefficients: (15 not defined because of singularities)
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -1986.3501   169.3349 -11.730   <2e-16
## Outlet_IdentifierOUT046           1959.9281    83.7867  23.392   <2e-16
## Outlet_IdentifierOUT045           1891.8411    83.5331  22.648   <2e-16
## Outlet_IdentifierOUT049           2057.6792    84.1093  24.464   <2e-16
## Outlet_IdentifierOUT035           2104.4092    83.8369  25.101   <2e-16
## Outlet_IdentifierOUT018           1683.1160    83.6592  20.119   <2e-16
## Outlet_IdentifierOUT019             12.0442    68.8349   0.175    0.861
## Outlet_IdentifierOUT027           3411.2793    84.0027  40.609   <2e-16
## Outlet_IdentifierOUT017           2062.8085    83.1515  24.808   <2e-16
## Outlet_IdentifierOUT013           1990.9033    83.5285  23.835   <2e-16
## Outlet_IdentifierOUT010                  NA         NA      NA       NA
## Outlet_TypeSupermarket.Type3             NA         NA      NA       NA
## Outlet_TypeSupermarket.Type2             NA         NA      NA       NA
## Outlet_TypeSupermarket.Type1             NA         NA      NA       NA
## Item_Type_CombinedFood              16.1708    43.9286   0.368    0.713
## Item_Type_CombinedNon.Consumable   -10.1285    49.0070  -0.207    0.836
```

```
## Outlet_TypeGrocery.Store                    NA         NA      NA       NA
## Item_Type_CombinedDrinks                    NA         NA      NA       NA
## Outlet_SizeSmall                            NA         NA      NA       NA
## Outlet_SizeMedium                           NA         NA      NA       NA
## Outlet_Location_TypeTier.2                  NA         NA      NA       NA
## Outlet_Location_TypeTier.3                  NA         NA      NA       NA
## Outlet_Location_TypeTier.1                  NA         NA      NA       NA
## Outlet_SizeHigh                             NA         NA      NA       NA
## Item_Fat_ContentRegular                40.7363    28.2782   1.441    0.150
## Item_Fat_ContentNon.Edible                  NA         NA      NA       NA
## Item_Fat_ContentLow.Fat                     NA         NA      NA       NA
## Outlet_Years                                NA         NA      NA       NA
## Item_Visibility_MeanRatio              71.1070    98.4560   0.722    0.470
## Item_MRP                               15.5588     0.1966  79.132   <2e-16
##
## (Intercept)                         ***
## Outlet_IdentifierOUT046             ***
## Outlet_IdentifierOUT045             ***
## Outlet_IdentifierOUT049             ***
## Outlet_IdentifierOUT035             ***
## Outlet_IdentifierOUT018             ***
## Outlet_IdentifierOUT019
## Outlet_IdentifierOUT027             ***
## Outlet_IdentifierOUT017             ***
## Outlet_IdentifierOUT013             ***
## Outlet_IdentifierOUT010
## Outlet_TypeSupermarket.Type3
## Outlet_TypeSupermarket.Type2
## Outlet_TypeSupermarket.Type1
## Item_Type_CombinedFood
## Item_Type_CombinedNon.Consumable
## Outlet_TypeGrocery.Store
## Item_Type_CombinedDrinks
## Outlet_SizeSmall
## Outlet_SizeMedium
## Outlet_Location_TypeTier.2
## Outlet_Location_TypeTier.3
## Outlet_Location_TypeTier.1
## Outlet_SizeHigh
## Item_Fat_ContentRegular
## Item_Fat_ContentNon.Edible
## Item_Fat_ContentLow.Fat
## Outlet_Years
## Item_Visibility_MeanRatio
## Item_MRP                            ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1128 on 8508 degrees of freedom
```

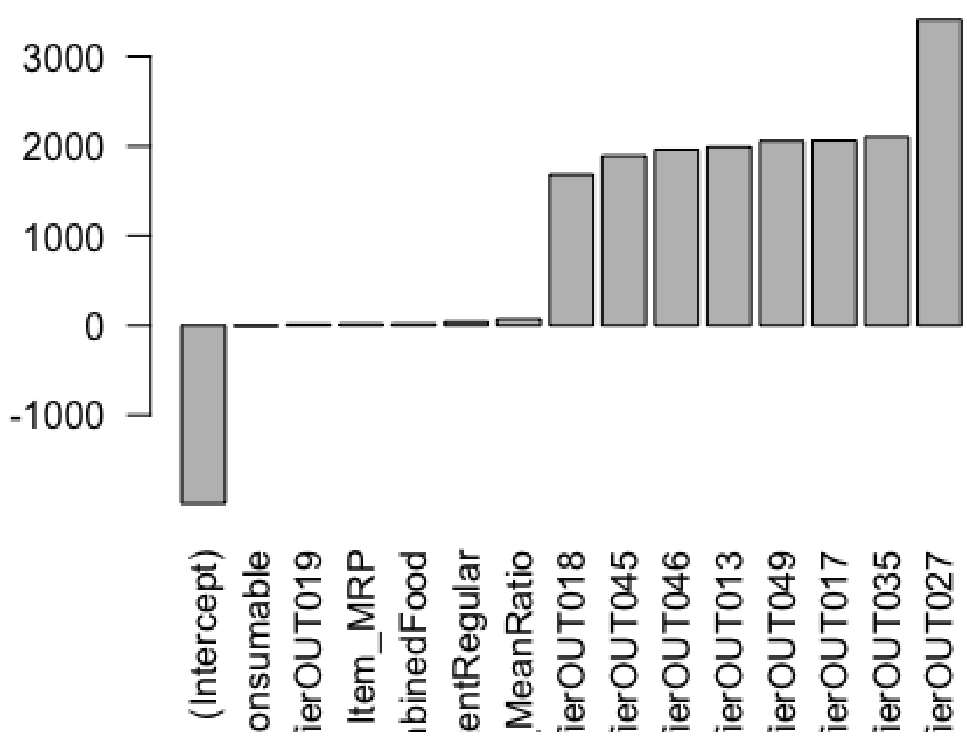
```
## Multiple R-squared:  0.5635, Adjusted R-squared:  0.5627
## F-statistic: 784.4 on 14 and 8508 DF,  p-value: < 2.2e-16
```

```r
barplot(sort(linear_model$coefficients), las=2)

linear_model <- lm(Item_Outlet_Sales ~ Outlet_IdentifierOUT046 + Outlet_Ident
ifierOUT045 + Outlet_IdentifierOUT049
                   + Outlet_IdentifierOUT035 + Outlet_IdentifierOUT018 + Outl
et_IdentifierOUT019 + Outlet_IdentifierOUT027
                   + Outlet_IdentifierOUT017 + Outlet_IdentifierOUT013 + Item
_Type_CombinedFood + Item_Type_CombinedNon.Consumable + Item_Fat_ContentRegul
ar + Item_Visibility_MeanRatio + Item_MRP, data = train)

summary(linear_model)
```

```
##
## Call:
## lm(formula = Item_Outlet_Sales ~ Outlet_IdentifierOUT046 + Outlet_Identifi
erOUT045 +
##     Outlet_IdentifierOUT049 + Outlet_IdentifierOUT035 + Outlet_IdentifierO
UT018 +
##     Outlet_IdentifierOUT019 + Outlet_IdentifierOUT027 + Outlet_IdentifierO
UT017 +
##     Outlet_IdentifierOUT013 + Item_Type_CombinedFood + Item_Type_CombinedN
on.Consumable +
##     Item_Fat_ContentRegular + Item_Visibility_MeanRatio + Item_MRP,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4331.8  -677.8   -88.9   572.5  7942.5
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -1986.3501   169.3349 -11.730   <2e-16
## Outlet_IdentifierOUT046           1959.9281    83.7867  23.392   <2e-16
## Outlet_IdentifierOUT045           1891.8411    83.5331  22.648   <2e-16
## Outlet_IdentifierOUT049           2057.6792    84.1093  24.464   <2e-16
## Outlet_IdentifierOUT035           2104.4092    83.8369  25.101   <2e-16
## Outlet_IdentifierOUT018           1683.1160    83.6592  20.119   <2e-16
## Outlet_IdentifierOUT019             12.0442    68.8349   0.175    0.861
## Outlet_IdentifierOUT027           3411.2793    84.0027  40.609   <2e-16
## Outlet_IdentifierOUT017           2062.8085    83.1515  24.808   <2e-16
## Outlet_IdentifierOUT013           1990.9033    83.5285  23.835   <2e-16
## Item_Type_CombinedFood              16.1708    43.9286   0.368    0.713
## Item_Type_CombinedNon.Consumable   -10.1285    49.0070  -0.207    0.836
## Item_Fat_ContentRegular             40.7363    28.2782   1.441    0.150
## Item_Visibility_MeanRatio           71.1070    98.4560   0.722    0.470
## Item_MRP                            15.5588     0.1966  79.132   <2e-16
##
```

```
## (Intercept)                         ***
## Outlet_IdentifierOUT046             ***
## Outlet_IdentifierOUT045             ***
## Outlet_IdentifierOUT049             ***
## Outlet_IdentifierOUT035             ***
## Outlet_IdentifierOUT018             ***
## Outlet_IdentifierOUT019
## Outlet_IdentifierOUT027             ***
## Outlet_IdentifierOUT017             ***
## Outlet_IdentifierOUT013             ***
## Item_Type_CombinedFood
## Item_Type_CombinedNon.Consumable
## Item_Fat_ContentRegular
## Item_Visibility_MeanRatio
## Item_MRP                            ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1128 on 8508 degrees of freedom
## Multiple R-squared:  0.5635, Adjusted R-squared:  0.5627
## F-statistic: 784.4 on 14 and 8508 DF,  p-value: < 2.2e-16
```

```
barplot(sort(linear_model$coefficients), las=2)
```



```
drop_columns <- c("X", "Item_Identifier", "Outlet_Identifier", "Item_Outlet_S
ales")
lm_test <- test[,!(names(test) %in% drop_columns)] #input_variables_values_tr
aining_datasets

predicted_sales_lm <- predict(linear_model, lm_test)
lm_test$Item_Outlet_Sales <- predicted_sales_lm
```

8.5. Comparison of Models

```
library(data.table)
library(caret)
train_control <- trainControl(method="repeatedcv", number=10, repeats=3)
lm_accuracy <- train(Item_Outlet_Sales ~ Outlet_IdentifierOUT046 + Outlet_Ide
ntifierOUT045 + Outlet_IdentifierOUT049
                     + Outlet_IdentifierOUT035 + Outlet_IdentifierOUT018 + Outl
et_IdentifierOUT019 + Outlet_IdentifierOUT027
                     + Outlet_IdentifierOUT017 + Outlet_IdentifierOUT013 + Item
_Type_CombinedFood + Item_Type_CombinedNon.Consumable + Item_Fat_ContentRegul
ar + Item_Visibility_MeanRatio + Item_MRP, data=train, trControl=train_contro
l, method="lm")
##LINEAR REGRESSION MODEL
print(lm_accuracy)

## Linear Regression
##
## 8523 samples
##   14 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 7671, 7671, 7670, 7671, 7671, 7672, ...
## Resampling results:
##
##   RMSE      Rsquared
##   1129.006  0.5626939
##
## Tuning parameter 'intercept' was held constant at a value of TRUE

dt_accuracy <- train(Item_Outlet_Sales ~ Outlet_IdentifierOUT046 + Outlet_Ide
ntifierOUT045 + Outlet_IdentifierOUT049
                     + Outlet_IdentifierOUT035 + Outlet_IdentifierOUT018 + Outl
et_IdentifierOUT019 + Outlet_IdentifierOUT027
                     + Outlet_IdentifierOUT017 + Outlet_IdentifierOUT013 + Outl
et_IdentifierOUT010 + Outlet_TypeSupermarket.Type3
                     + Outlet_TypeSupermarket.Type2 + Outlet_TypeSupermarket.Ty
pe1 + Item_Type_CombinedFood +
                       Item_Type_CombinedNon.Consumable + Outlet_TypeGrocery.St
ore + Item_Type_CombinedDrinks + Outlet_SizeSmall
                     + Outlet_SizeMedium + Outlet_Location_TypeTier.2 + Outlet_
Location_TypeTier.3 + Outlet_Location_TypeTier.1
```

```
                   + Outlet_SizeHigh + Item_Fat_ContentRegular + Item_Fat_Con
tentNon.Edible + Item_Fat_ContentLow.Fat
                   + Outlet_Years + Item_Visibility_MeanRatio + Item_MRP, dat
a = train, method = "rpart", trControl=train_control)
##DECISION TREE
print(dt_accuracy)
```

```
## CART
##
## 8523 samples
##   29 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 7671, 7670, 7671, 7670, 7671, 7671, ...
## Resampling results across tuning parameters:
##
##   cp          RMSE      Rsquared
##   0.05892632  1296.950  0.4223624
##   0.16317858  1406.671  0.3181780
##   0.23662590  1598.796  0.2206137
##
## RMSE was used to select the optimal model using  the smallest value.
## The final value used for the model was cp = 0.05892632.
```

```
##RANDOM FOREST
which.min(rf$mse)
```

```
## [1] 990
```

Inferences:

Based on the model comparison we can see that Random Forests outperform Decision Trees and Linear Regression Models. This is because of the optimal selection of the parameters and the dependent variables.

To make the model better:

We can try several sets of parameters to identify the optimal set of predictors. With these predictors, we can make use of the RandomForest model.