BigMart

Ankita Patidar

12/1/2019

library(data.table)

## Warning: package 'data.table' was built under R version 3.6.2

#used for reading and manipulation  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.2

#used for plotting  
library(corrplot)

## Warning: package 'corrplot' was built under R version 3.6.2

## corrplot 0.84 loaded

#used for correlation plotting  
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.6.2

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':  
##   
## between, first, last

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

#used for manipulation and joining  
library(cowplot)

## Warning: package 'cowplot' was built under R version 3.6.2

##   
## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Note: As of version 1.0.0, cowplot does not change the

## default ggplot2 theme anymore. To recover the previous

## behavior, execute:  
## theme\_set(theme\_cowplot())

## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# used for combining multiple plots  
library(caret)

## Warning: package 'caret' was built under R version 3.6.2

## Loading required package: lattice

# used for modeling  
library(xgboost)

## Warning: package 'xgboost' was built under R version 3.6.2

##   
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':  
##   
## slice

# used for building XGBoost model

##install.packages("cowplot")  
##install.packages("caret")  
##install.packages("xgboost")  
#install.packages("data.table")

## pre-processing

#read the data  
train = fread("BigMart\_Train.csv")  
test = fread("BigMart\_Test.csv")  
submission = fread("big\_Mart\_Submission.csv", skip = 1)

#check the no. of columns and rows in train and test data  
dim(train)

## [1] 8523 12

dim(test)

## [1] 5681 11

#features of train and test data  
names(train)

## [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"

names(test)

## [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"

Train dataset has 8523 rows and 12 features and test has 5681 rows and 11 columns. train has 1 extra column which is the target variable.

#brief summary of the features of train and test data  
str(train)

## Classes 'data.table' and 'data.frame': 8523 obs. of 12 variables:  
## $ Item\_Identifier : chr "FDA15" "DRC01" "FDN15" "FDX07" ...  
## $ Item\_Weight : num 9.3 5.92 17.5 19.2 8.93 ...  
## $ Item\_Fat\_Content : chr "Low Fat" "Regular" "Low Fat" "Regular" ...  
## $ Item\_Visibility : num 0.016 0.0193 0.0168 0 0 ...  
## $ Item\_Type : chr "Dairy" "Soft Drinks" "Meat" "Fruits and Vegetables" ...  
## $ Item\_MRP : num 249.8 48.3 141.6 182.1 53.9 ...  
## $ Outlet\_Identifier : chr "OUT049" "OUT018" "OUT049" "OUT010" ...  
## $ Outlet\_Establishment\_Year: int 1999 2009 1999 1998 1987 2009 1987 1985 2002 2007 ...  
## $ Outlet\_Size : chr "Medium" "Medium" "Medium" "" ...  
## $ Outlet\_Location\_Type : chr "Tier 1" "Tier 3" "Tier 1" "Tier 3" ...  
## $ Outlet\_Type : chr "Supermarket Type1" "Supermarket Type2" "Supermarket Type1" "Grocery Store" ...  
## $ Item\_Outlet\_Sales : num 3735 443 2097 732 995 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

str(test)

## Classes 'data.table' and 'data.frame': 5681 obs. of 11 variables:  
## $ Item\_Identifier : chr "FDW58" "FDW14" "NCN55" "FDQ58" ...  
## $ Item\_Weight : num 20.75 8.3 14.6 7.32 NA ...  
## $ Item\_Fat\_Content : chr "Low Fat" "reg" "Low Fat" "Low Fat" ...  
## $ Item\_Visibility : num 0.00756 0.03843 0.09957 0.01539 0.1186 ...  
## $ Item\_Type : chr "Snack Foods" "Dairy" "Others" "Snack Foods" ...  
## $ Item\_MRP : num 107.9 87.3 241.8 155 234.2 ...  
## $ Outlet\_Identifier : chr "OUT049" "OUT017" "OUT010" "OUT017" ...  
## $ Outlet\_Establishment\_Year: int 1999 2007 1998 2007 1985 1997 2009 1985 2002 2007 ...  
## $ Outlet\_Size : chr "Medium" "" "" "" ...  
## $ Outlet\_Location\_Type : chr "Tier 1" "Tier 2" "Tier 3" "Tier 2" ...  
## $ Outlet\_Type : chr "Supermarket Type1" "Supermarket Type1" "Grocery Store" "Supermarket Type1" ...  
## - attr(\*, ".internal.selfref")=<externalptr>

#combine train and test to perform modifcation on the data  
test[,"Item\_Outlet\_Sales" := NA]  
combi = rbind(train,test)  
dim(combi)

## [1] 14204 12

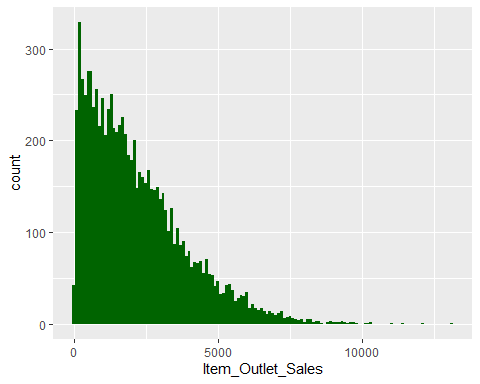
str(combi)

## Classes 'data.table' and 'data.frame': 14204 obs. of 12 variables:  
## $ Item\_Identifier : chr "FDA15" "DRC01" "FDN15" "FDX07" ...  
## $ Item\_Weight : num 9.3 5.92 17.5 19.2 8.93 ...  
## $ Item\_Fat\_Content : chr "Low Fat" "Regular" "Low Fat" "Regular" ...  
## $ Item\_Visibility : num 0.016 0.0193 0.0168 0 0 ...  
## $ Item\_Type : chr "Dairy" "Soft Drinks" "Meat" "Fruits and Vegetables" ...  
## $ Item\_MRP : num 249.8 48.3 141.6 182.1 53.9 ...  
## $ Outlet\_Identifier : chr "OUT049" "OUT018" "OUT049" "OUT010" ...  
## $ Outlet\_Establishment\_Year: int 1999 2009 1999 1998 1987 2009 1987 1985 2002 2007 ...  
## $ Outlet\_Size : chr "Medium" "Medium" "Medium" "" ...  
## $ Outlet\_Location\_Type : chr "Tier 1" "Tier 3" "Tier 1" "Tier 3" ...  
## $ Outlet\_Type : chr "Supermarket Type1" "Supermarket Type2" "Supermarket Type1" "Grocery Store" ...  
## $ Item\_Outlet\_Sales : num 3735 443 2097 732 995 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

## Item\_MRP, Item\_Visibility, Item\_Outlet\_Sales,Item\_Weight is continous variables

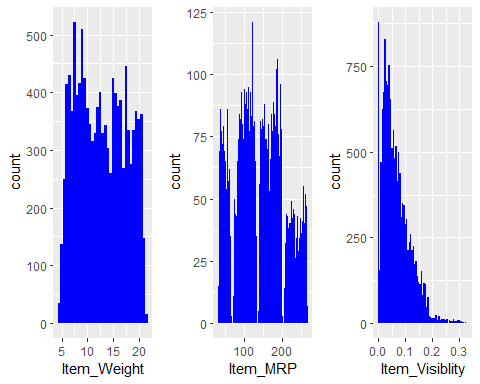
Visualize the continuous variables using histograms and categorical variables using bar plots.

#Plot target variable "Item\_Outlet\_Sales"  
  
ggplot(data = train) + geom\_histogram(aes(train$Item\_Outlet\_Sales), binwidth = 100, fill = "darkgreen") + xlab("Item\_Outlet\_Sales")

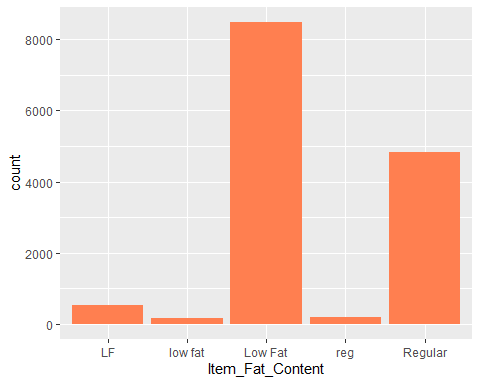


# Target variable is right skewed  
  
#Check the distribution of other numeric independent variable  
p1 = ggplot(combi) + geom\_histogram(aes(combi$Item\_Weight), binwidth = 0.5, fill = "blue") + xlab("Item\_Weight")  
#No clear pattern in Item\_Weight  
  
p2 = ggplot(combi) + geom\_histogram(aes(combi$Item\_MRP), binwidth = 1, fill = "blue") + xlab("Item\_MRP")  
#4 different distributions for Item\_MRP  
p3 = ggplot(combi) + geom\_histogram(aes(combi$Item\_Visibility), binwidth = 0.005, fill = "blue") + xlab("Item\_Visiblity")  
#Item\_Visibility is right-skewed  
  
#plot\_grid is used to combine plots  
plot\_grid(p1, p2,p3, nrow = 1)

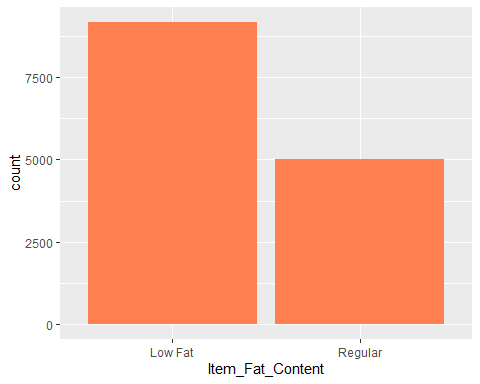
## Warning: Removed 2439 rows containing non-finite values (stat\_bin).

 ##Outlet\_Location\_Type,Outlet\_Type, Outlet\_Identifier,Item\_Type,Item\_Identifier,Item\_Fat\_Content,Outlet\_Size,

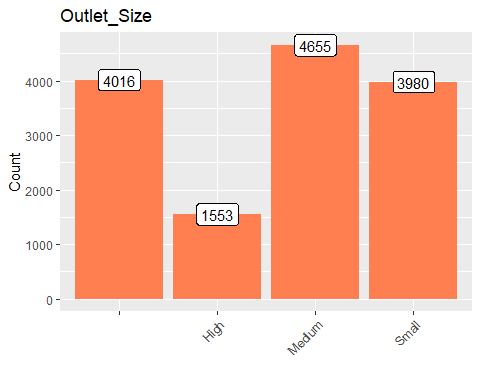
#Plot for independent categorical variables  
  
## Item\_Fat\_Content  
ggplot(combi) + geom\_bar(aes(combi$Item\_Fat\_Content), fill = "coral") + xlab("Item\_Fat\_Content")



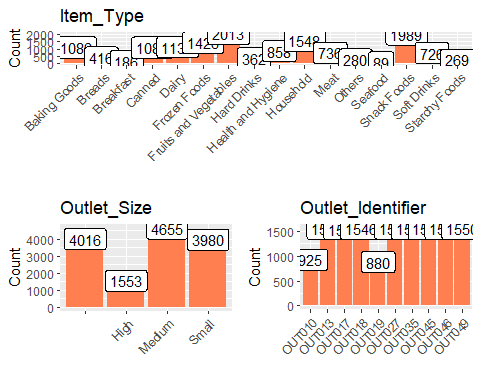
#Combining LF, low fat, Low Fat into one Low Fat  
combi$Item\_Fat\_Content[combi$Item\_Fat\_Content == "LF"] = "Low Fat"  
combi$Item\_Fat\_Content[combi$Item\_Fat\_Content == "low fat"] = "Low Fat"  
#combining reg and Regular into one Regular  
combi$Item\_Fat\_Content[combi$Item\_Fat\_Content == "reg"] = "Regular"  
ggplot(combi) + geom\_bar(aes(combi$Item\_Fat\_Content), fill = "coral") + xlab("Item\_Fat\_Content")



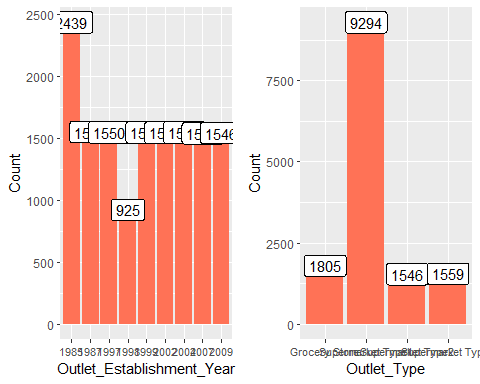
p4 = ggplot(combi %>% group\_by(Item\_Type) %>% summarise(Count =n())) + geom\_bar(aes(Item\_Type, Count), stat = "identity", fill = "coral") + xlab("") + geom\_label(aes(Item\_Type, Count, label = Count), vjust = 0.5) + theme(axis.text.x = element\_text(angle= 45, hjust = 1)) + ggtitle("Item\_Type")  
  
  
p5 = ggplot(combi %>% group\_by(Item\_Identifier) %>% summarise(Count =n())) + geom\_bar(aes(Item\_Identifier, Count), stat = "identity", fill = "coral") + xlab("") + geom\_label(aes(Item\_Identifier, Count, label = Count), vjust = 0.5) + theme(axis.text.x = element\_text(angle= 45, hjust = 1)) + ggtitle("Item\_Identifier")  
  
  
p6 = ggplot(combi %>% group\_by(Outlet\_Size) %>% summarise(Count =n())) + geom\_bar(aes(Outlet\_Size, Count), stat = "identity", fill = "coral") + xlab("") + geom\_label(aes(Outlet\_Size, Count, label = Count), vjust = 0.5) + theme(axis.text.x = element\_text(angle= 45, hjust = 1)) + ggtitle("Outlet\_Size")  
p6



p7 = ggplot(combi %>% group\_by(Outlet\_Identifier) %>% summarise(Count =n())) + geom\_bar(aes(Outlet\_Identifier, Count), stat = "identity", fill = "coral") + xlab("") + geom\_label(aes(Outlet\_Identifier, Count, label = Count), vjust = 0.5) + theme(axis.text.x = element\_text(angle= 45, hjust = 1)) + ggtitle("Outlet\_Identifier")  
  
second\_row = plot\_grid( p6, p7, nrow = 1)  
  
plot\_grid(p4, second\_row, ncol = 1)

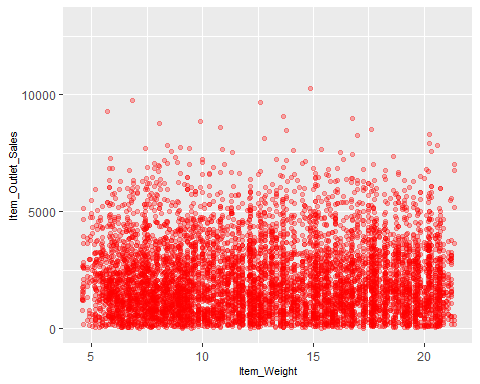


p8 = ggplot(combi %>% group\_by(Outlet\_Establishment\_Year) %>% summarise(Count = n())) + geom\_bar(aes(factor(Outlet\_Establishment\_Year), Count), stat = "identity", fill = "coral1") + geom\_label(aes(factor(Outlet\_Establishment\_Year), Count, label = Count), vjust = 0.5) + xlab("Outlet\_Establishment\_Year") + theme(axis.text.x = element\_text(size = 8.5))  
#1998 has lesser number of observation  
  
  
p9 = ggplot(combi %>% group\_by(Outlet\_Type) %>% summarise(Count = n())) + geom\_bar(aes(Outlet\_Type, Count), stat = "identity", fill = "coral1") + geom\_label(aes(factor(Outlet\_Type), Count, label = Count), vjust = 0.5) + theme(axis.text.x = element\_text(size = 8.5))  
#Supermarket Type 1 seems to be most popular  
  
plot\_grid(p8, p9, ncol = 2)

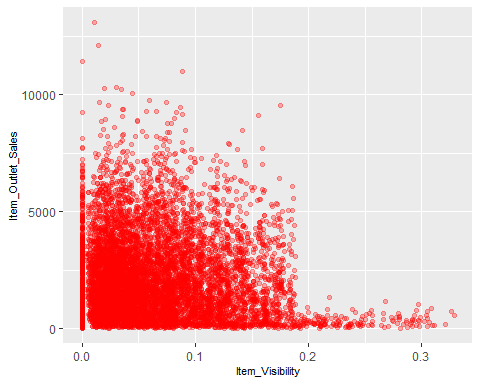


#bi-variate analysis - explore the independent variables with respect to the target variable.  
#use of scatter plots for the continuous or numeric variables and violin plots for the categorical variables.  
# extracting train data from the combined data'  
  
train = combi[1:nrow(train)]  
  
# Item\_Weight vs Item\_Outlet\_Sales  
  
p10 = ggplot(train) + geom\_point(aes(Item\_Weight, Item\_Outlet\_Sales), colour = "red", alpha = 0.3) +theme(axis.title = element\_text(size=8.5))  
p10

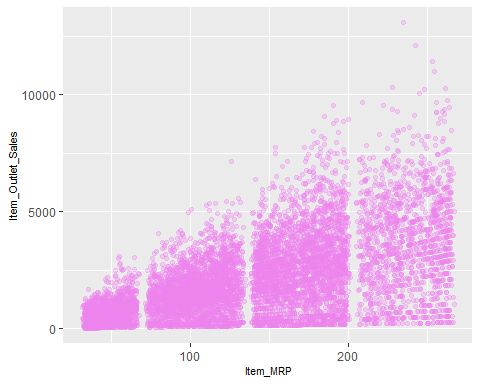
## Warning: Removed 1463 rows containing missing values (geom\_point).



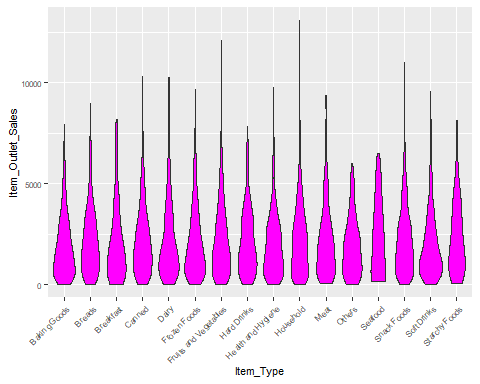
# Item\_Visibility vs Item\_Outlet\_Sales  
  
p11 = ggplot(train) + geom\_point(aes(Item\_Visibility,Item\_Outlet\_Sales), colour = "red", alpha = 0.3) + theme(axis.title = element\_text(size = 8.5))  
p11



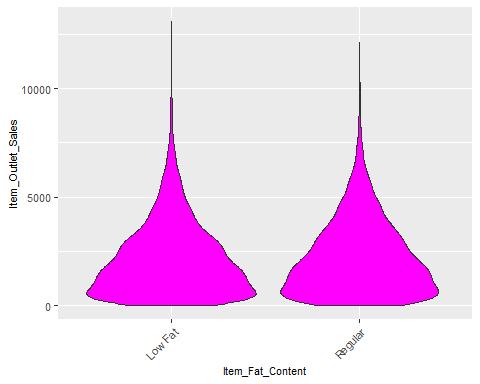
#there is a string of points at Item\_Visibility = 0.0 which seems strange as item visibility cannot be completely zero  
  
# Item\_MRP vs Item\_Outlet\_Sales  
  
p12 = ggplot(train) + geom\_point(aes(Item\_MRP, Item\_Outlet\_Sales), colour = "violet", alpha = 0.3) + theme(axis.title = element\_text(size = 8.5))  
p12



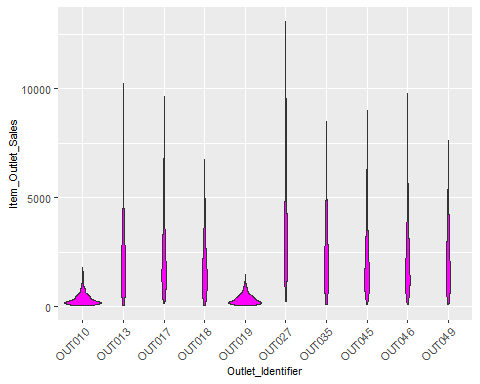
# Item\_Type vs Item\_Outlet\_Sales   
  
p13 = ggplot(train) + geom\_violin(aes(Item\_Type,Item\_Outlet\_Sales), fill = "Magenta") + theme(axis.text.x = element\_text(angle = 45, hjust = 1), axis.text = element\_text(size = 6), axis.title = element\_text(size = 8.5))  
p13



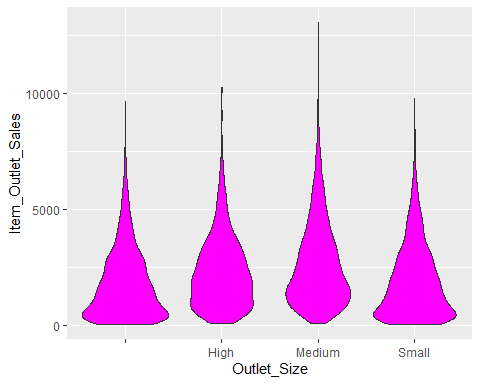
# Item\_Fat\_Content vs Item\_Outlet\_Sales  
p14 = ggplot(train) + geom\_violin(aes(Item\_Fat\_Content, Item\_Outlet\_Sales), fill = "magenta") + theme(axis.text.x = element\_text(angle = 45, hjust = 1), axis.text = element\_text(size = 8), axis.title = element\_text(size = 8.5))  
  
 p14



# Outlet\_Identifier vs Item\_Outlet\_Sales  
  
p15= ggplot(train) + geom\_violin(aes(Outlet\_Identifier, Item\_Outlet\_Sales), fill = "magenta") + theme(axis.text.x = element\_text(angle = 45, hjust = 1), axis.text = element\_text(size = 8), axis.title = element\_text(size = 8.5))  
  
p15

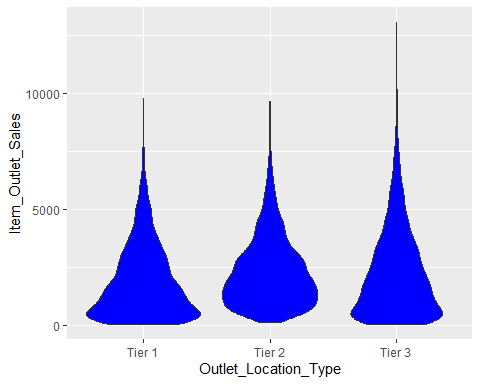


p16 = ggplot(train) + geom\_violin(aes(Outlet\_Size, Item\_Outlet\_Sales), fill = "magenta")  
  
p16

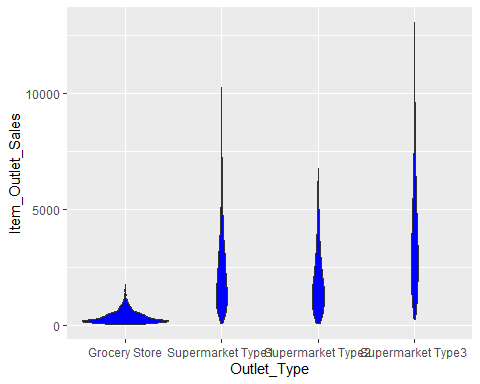


#The distribution of ‘Small’ Outlet\_Size is almost identical to the distribution of the blank category (first vioin) of Outlet\_Size. So, we can substitute the blanks in Outlet\_Size with ‘Small’.

p17 = ggplot(train) + geom\_violin(aes(Outlet\_Location\_Type, Item\_Outlet\_Sales), fill = "blue")  
#Tier 1 and Tier 3 locations of Outlet\_Location\_Type look similar.  
p18 = ggplot(train) + geom\_violin(aes(Outlet\_Type, Item\_Outlet\_Sales), fill = "blue")  
#Grocery Store has most of its data points around the lower sales  
  
p17



p18



#finding missing values in variables  
sum(is.na(combi$Item\_Weight))

## [1] 2439

#2439 missing values in Item\_Weight  
sum(is.na(combi$Item\_Fat\_Content))

## [1] 0

sum(is.na(combi$Item\_Identifier))

## [1] 0

sum(is.na(combi$Item\_Visibility))

## [1] 0

sum(is.na(combi$Item\_Type))

## [1] 0

sum(is.na(combi$Item\_MRP))

## [1] 0

sum(is.na(combi$Outlet\_Identifier))

## [1] 0

sum(is.na(combi$Outlet\_Establishment\_Year))

## [1] 0

sum(is.na(combi$Outlet\_Size))

## [1] 0

sum(is.na(combi$Outlet\_Location\_Type))

## [1] 0

sum(is.na(combi$Outlet\_Type))

## [1] 0

sum(is.na(combi$Item\_Outlet\_Sales))

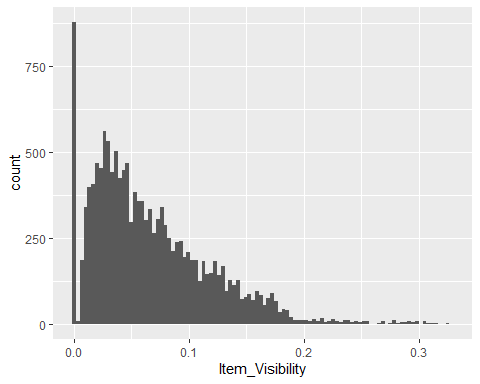
## [1] 5681

#5681 missing values in Item\_Outlet\_Sales

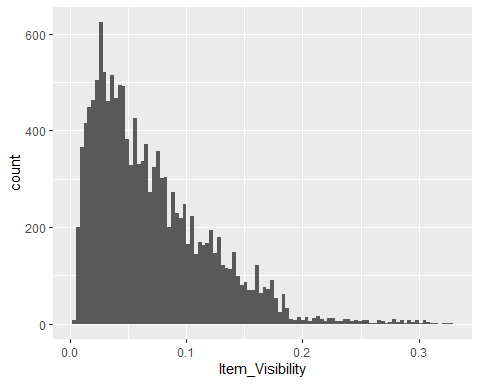
#impute Item\_Weight with mean weight based on the Item\_Identifier variable  
missing\_index = which(is.na(combi$Item\_Weight))  
for (i in missing\_index){  
 item = combi$Item\_Identifier[i]  
 combi$Item\_Weight[i] = mean(combi$Item\_Weight[combi$Item\_Identifier == item], na.rm = T)  
}  
  
sum(is.na(combi$Item\_Weight))

## [1] 0

ggplot(combi) + geom\_histogram(aes(Item\_Visibility), bins =100)



zero\_index = which(combi$Item\_Visibility == 0)  
#zeroes in Item\_Visibility variable can be replaced with Item\_Identifier wise mean values of Item\_Visibility  
for (i in zero\_index){  
 item = combi$Item\_Identifier[i]  
 combi$Item\_Visibility[i] = mean(combi$Item\_Visibility[combi$Item\_Identifier==item], na.rm = T)  
}  
  
ggplot(combi) + geom\_histogram(aes(Item\_Visibility), bins = 100)



#the issue of zero item visibility has been resolved.

Create new features to help improving the model performance:

Item\_Type\_new: Broader categories for the variable Item\_Type. Item\_category: Categorical variable derived from Item\_Identifier. Outlet\_Years: Years of operation for outlets. price\_per\_unit\_wt: Item\_MRP/Item\_Weight Item\_MRP\_clusters: Binned feature for Item\_MRP.

#the Item\_Type variable can be classified into the categories perishable and non\_perishable  
perishable = c("Breads", "Breakfast", "Dairy", "Fruits and Vegetables", "Meat", "Seafood")  
  
non\_perishable = c("Baking Goods", "Canned", "Frozen Foods", "Hard Drinks", "Health and Hygiene", "Household", "Soft Drinks")  
  
# create a new feature 'Item\_Type\_new'   
  
combi[, Item\_Type\_New := ifelse(Item\_Type %in% perishable, "perishable", ifelse(Item\_Type %in% non\_perishable, "non\_perishable", "not sure"))]

#the first 2 characters of Item\_Identifier, i.e., ‘DR’, ‘FD’, and ‘NC’ stand for drinks, food, and non-consumable.  
table(combi$Item\_Type, substr(combi$Item\_Identifier,1,2))

##   
## DR FD NC  
## Baking Goods 0 1086 0  
## Breads 0 416 0  
## Breakfast 0 186 0  
## Canned 0 1084 0  
## Dairy 229 907 0  
## Frozen Foods 0 1426 0  
## Fruits and Vegetables 0 2013 0  
## Hard Drinks 362 0 0  
## Health and Hygiene 0 0 858  
## Household 0 0 1548  
## Meat 0 736 0  
## Others 0 0 280  
## Seafood 0 89 0  
## Snack Foods 0 1989 0  
## Soft Drinks 726 0 0  
## Starchy Foods 0 269 0

combi[, Item\_Category := substr(combi$Item\_Identifier,1,2)]  
  
combi$Item\_Fat\_Content[combi$Item\_Category == "NC"] = "non-edible"  
  
  
# years of operation for outlets   
combi[, Item\_Years := 2013 - combi$Outlet\_Establishment\_Year]  
combi$Outlet\_Establishment\_Year = as.factor(combi$Outlet\_Establishment\_Year)  
  
# Price per unit weight   
  
combi[, PPUW := Item\_MRP/Item\_Weight]  
  
# creating new independent variable - Item\_MRP\_clusters  
combi[,Item\_MRP\_Clusters := ifelse(Item\_MRP < 69, "1st", ifelse(Item\_MRP >= 69 & Item\_MRP < 139, "2nd", ifelse(Item\_MRP >= 139 & Item\_MRP < 203, "3rd", "4th")))]

#Label encode Outlet\_Size and Outlet\_Location\_Type as these are ordinal variables.  
combi[,Outlet\_Size\_Num := ifelse(Outlet\_Size == "Small", 0, ifelse(Outlet\_Size == "Medium", 1,2))]  
  
combi[,Outlet\_Location\_Type\_num := ifelse(Outlet\_Location\_Type == "Tier 3", 0, ifelse(Outlet\_Location\_Type == "Tier 2", 1, 2))]

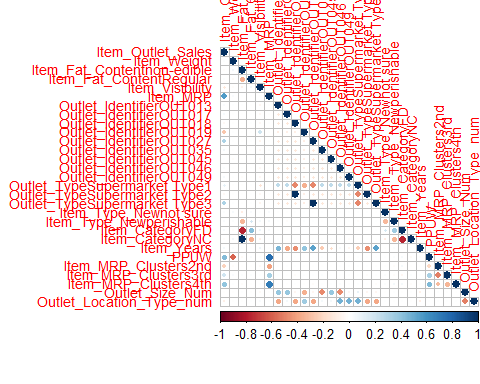
# removing categorical variables after label encoding   
  
combi[,c("Outlet\_Size", "Outlet\_Location\_Type") := NULL]

#One hot encoding for the categorical variable  
ohe = dummyVars("~." , data = combi[, -c("Item\_Identifier", "Outlet\_Establishment\_Year", "Item\_Type")], fullRank = T)  
  
ohe\_df = data.table(predict(ohe, combi[,-c("Item\_Identifier", "Outlet\_Establishment\_Year", "Item\_Type")]))  
  
combi = cbind(combi[,"Item\_Identifier"], ohe\_df)

combi[,Item\_Visibility := log(Item\_Visibility + 1)]  
# log + 1 to avoid division by zero  
combi[,PPUW := log(PPUW + 1)]

#scale and center the numeric variables to make them have a mean of zero, standard deviation of one and scale of 0 to 1.  
  
# index of numeric features  
num\_vars = which(sapply(combi, is.numeric))  
  
num\_vars\_names = names(num\_vars)  
combi\_numeric = combi[,setdiff(num\_vars\_names, "Item\_Outlet\_Sales"), with= F]  
prep\_num = preProcess(combi\_numeric, method = c("center", "scale"))  
combi\_numeric\_norm = predict(prep\_num, combi\_numeric)  
  
 # removing numeric independent   
combi[,setdiff(num\_vars\_names, "Item\_Outlet\_Sales") := NULL]   
  
combi = cbind(combi, combi\_numeric\_norm)  
  
#Splittng  
  
train = combi[1:nrow(train)]   
test = combi[(nrow(train) + 1):nrow(combi)]  
test[,Item\_Outlet\_Sales := NULL]   
# removing Item\_Outlet\_Sales as it contains only NA for test dataset

cor\_train = cor(train[,-c("Item\_Identifier")])  
corrplot(cor\_train, method = "circle", type = "lower", tl.cex = 0.9)



#Regression Model  
linear\_reg\_model = lm(Item\_Outlet\_Sales ~., data =train[,-c("Item\_Identifier")])  
linear\_reg\_model

##   
## Call:  
## lm(formula = Item\_Outlet\_Sales ~ ., data = train[, -c("Item\_Identifier")])  
##   
## Coefficients:  
## (Intercept) Item\_Weight   
## 2181.357 -45.843   
## `Item\_Fat\_Contentnon-edible` Item\_Fat\_ContentRegular   
## -1.784 19.508   
## Item\_Visibility Item\_MRP   
## -10.864 985.536   
## Outlet\_IdentifierOUT013 Outlet\_IdentifierOUT017   
## 605.756 626.660   
## Outlet\_IdentifierOUT018 Outlet\_IdentifierOUT019   
## 509.250 4.126   
## Outlet\_IdentifierOUT027 Outlet\_IdentifierOUT035   
## 1050.623 640.765   
## Outlet\_IdentifierOUT045 Outlet\_IdentifierOUT046   
## 574.065 595.725   
## Outlet\_IdentifierOUT049 `Outlet\_TypeSupermarket Type1`   
## 625.962 NA   
## `Outlet\_TypeSupermarket Type2` `Outlet\_TypeSupermarket Type3`   
## NA NA   
## `Item\_Type\_Newnot sure` Item\_Type\_Newperishable   
## -1.012 6.861   
## Item\_CategoryFD Item\_CategoryNC   
## 7.884 NA   
## Item\_Years PPUW   
## NA -74.728   
## Item\_MRP\_Clusters2nd Item\_MRP\_Clusters3rd   
## 27.393 55.944   
## Item\_MRP\_Clusters4th Outlet\_Size\_Num   
## 43.687 NA   
## Outlet\_Location\_Type\_num   
## NA

# preparing dataframe for submission and writing it in a csv file   
submission$Item\_Outlet\_Sales = predict(linear\_reg\_model, test[,-c("Item\_Identifier")])

## Warning in predict.lm(linear\_reg\_model, test[, -c("Item\_Identifier")]):  
## prediction from a rank-deficient fit may be misleading

write.csv(submission, "Linear\_Reg\_submit.csv", row.names = F)