DineSafe Food Hygiene Dataset Analysis

Mohammed Amir

March 01, 2017

## Load Dataset

## Convert NULL values to NA on data loading  
Dinesafe = read.csv("D:/CAPSTONE/data/DineSafe\_02162017.csv", na.strings='NULL')  
Address = read.csv("D:/CAPSTONE/data/ADDRESS\_02262017.csv", na.strings='NULL')

# Identify column names

colnames(Dinesafe)

## Review dimension of dataset (Row by column)

dim(Dinesafe)  
dim(Address)

## Review summary of dinesafe dataset

summary(Dinesafe)

## Review summary of Address dataset

summary(Address)

## Review structure of Dinesafe dataset

str(Dinesafe)

## Review structure of Address dataset

str(Address)

### Merge Dinesafe and Address dataset based on establishment id column

Dinesafe <- merge(Dinesafe,Address,by="ESTABLISHMENT\_ID")

## Remove Columns from data set

### Remove COURT\_OUTCOME and AMOUNT\_FINED Columns from Dinesafe dataset  
Dinesafe <- subset(Dinesafe, select = -c(ROW\_ID, COURT\_OUTCOME,AMOUNT\_FINED,LONG\_ADDRESS) )  
  
### Remove Establishement Name from Address dataset to avoid duplicate rows   
Dinesafe <- subset(Dinesafe, select = -c(ESTABLISHMENT\_NAME.y, ESTABLISHMENT\_ADDRESS) )  
  
## Rename ESTABLISHMENT\_NAME.x column name to ESTABLISHMENT\_NAME  
colnames(Dinesafe)[colnames(Dinesafe) == 'ESTABLISHMENT\_NAME.x'] <- 'ESTABLISHMENT\_NAME'

## Convert VALUE from factor to numeric Data Structure

## use suppressWarnings to avoid getting "NAs introduced by coercion" message  
Dinesafe$VALUE <- suppressWarnings(as.numeric(as.character(Dinesafe$VALUE)))

## Convert INSPECTION\_DATE from factor to Date Data Structure

Dinesafe$INSPECTION\_DATE <- dmy(as.character(Dinesafe$INSPECTION\_DATE))

## Convert INSPECTION\_DATE from factor to Date Data Structure

Dinesafe$CUISINE\_TYPE <- as.character(Dinesafe$CUISINE\_TYPE)  
Dinesafe$ESTABLISHMENT\_TYPE <- as.character(Dinesafe$ESTABLISHMENT\_TYPE)  
Dinesafe$ESTABLISHMENT\_STATUS <- as.character(Dinesafe$ESTABLISHMENT\_STATUS)   
Dinesafe$SEVERITY <- as.character(Dinesafe$SEVERITY)  
Dinesafe$ACTION <- as.character(Dinesafe$ACTION)  
Dinesafe$ACTION <- as.character(Dinesafe$ACTION)  
Dinesafe$ACTION <- as.character(Dinesafe$ACTION)

str(Dinesafe)

## Complete Case Rows with no missing (NA) value

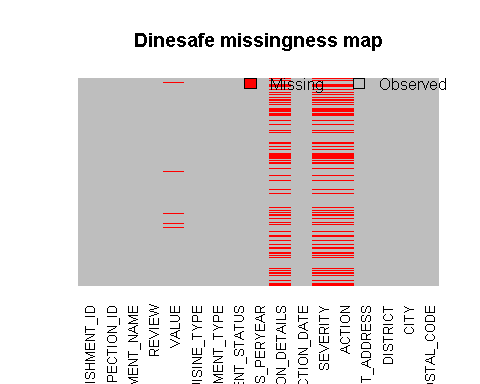
Complete\_Dinesafe <- Dinesafe[complete.cases(Dinesafe),]  
nrow(Complete\_Dinesafe)

## Rows with Missing (NA) value

Dinesafe\_NA <- Dinesafe[!complete.cases(Dinesafe),]  
nrow(Dinesafe\_NA)

## Show missingness in Dinesafe dataset

#Quantify missing values  
apply(Dinesafe, 2, function(x) sum(is.na(x)))  
   
# Plot missingness map  
missmap(Dinesafe, col = c("Red","Grey"), y.cex = 0.8, x.cex = 0.8, legend = TRUE, rank.order = "False" ,main = "Dinesafe missingness map", y.labels = NULL,y.at = NULL)



## Impute Dinesafe$REVIEW with Mean Review Value for each missing review value based cuisine type

Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="African"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="African"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Bakeries"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Bakeries"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Bar"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Bar"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Cafe"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Cafe"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Caribbean"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Caribbean"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Deli"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Deli"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Dessert"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Dessert"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="European"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="European"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Far Eastern"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Far Eastern"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Pastries"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Pastries"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="South Asian"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="South Asian"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="South East Asian"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="South East Asian"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Latin American"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Latin American"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Mediterranean"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Mediterranean"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Middle Eastern"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Middle Eastern"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="North American"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="North American"], na.rm=TRUE)  
Dinesafe$REVIEW[is.na(Dinesafe$REVIEW) & Dinesafe$CUISINE\_TYPE=="Juicery & Smoothies"] = mean(Dinesafe$REVIEW[Dinesafe$CUISINE\_TYPE=="Juicery & Smoothies"], na.rm=TRUE)

## Impute Dinesafe$VALUE with Mean Value for each missing value based cuisine type

Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="African"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="African"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Bakeries"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Bakeries"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Bar"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Bar"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Cafe"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Cafe"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Caribbean"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Caribbean"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Deli"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Deli"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Dessert"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Dessert"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="European"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="European"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Far Eastern"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Far Eastern"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Pastries"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Pastries"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="South Asian"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="South Asian"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="South East Asian"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="South East Asian"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Latin American"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Latin American"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Mediterranean"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Mediterranean"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Middle Eastern"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Middle Eastern"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="North American"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="North American"], na.rm=TRUE)  
Dinesafe$VALUE[is.na(Dinesafe$VALUE) & Dinesafe$CUISINE\_TYPE=="Juicery & Smoothies"] = mean(Dinesafe$VALUE[Dinesafe$CUISINE\_TYPE=="Juicery & Smoothies"], na.rm=TRUE)

## Impute Severity column if it is NA and Establishment Status is PASS  
Dinesafe$SEVERITY[is.na(Dinesafe$SEVERITY) & Dinesafe$ESTABLISHMENT\_STATUS == "Pass"] = "N - No Action"  
  
## Impute Action column if it is NA and Establishment Status is PASS & Severity is No Action  
Dinesafe$ACTION[is.na(Dinesafe$ACTION) & Dinesafe$ESTABLISHMENT\_STATUS == "Pass" & Dinesafe$SEVERITY == "N - No Action"] = "No Action Required"

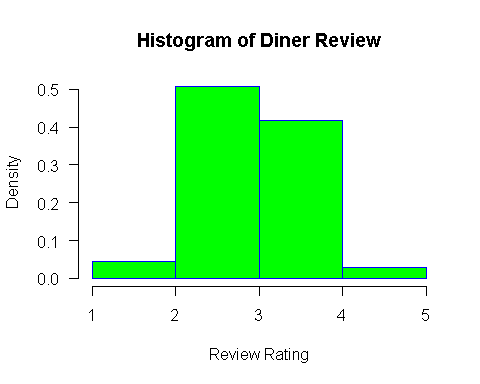
## Double Check Rows with Missing (NA) value after imputation

Complete\_Dinesafe\_1 <- Dinesafe[complete.cases(Dinesafe),]  
Complete\_Dinesafe\_1  
nrow(Complete\_Dinesafe\_1)  
  
Dinesafe\_NA1 <- Dinesafe[!complete.cases(Dinesafe),]  
Dinesafe\_NA1  
nrow(Dinesafe\_NA1)

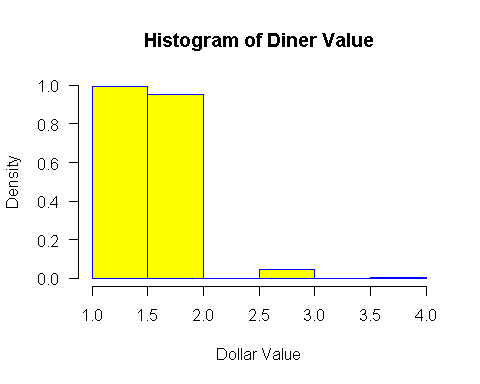
### 

### Univariate Analysis & Visualization

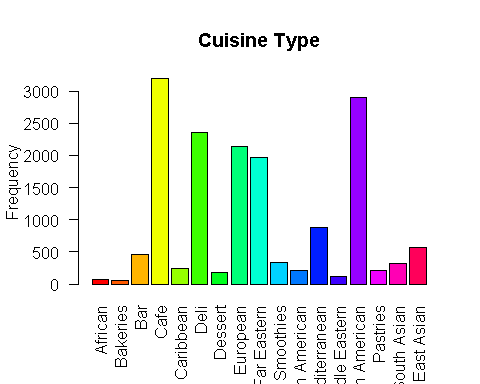
hist(Dinesafe$REVIEW, xlab = "Review Rating", main="Histogram of Diner Review",   
 border="blue", col="GREEN", las=1, breaks=5,prob = TRUE)



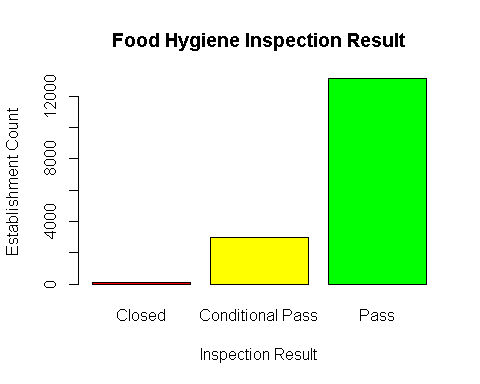
hist(Dinesafe$VALUE, xlab = "Dollar Value", main="Histogram of Diner Value",   
 border="blue", col="YELLOW", las=1, breaks=5,prob = TRUE)



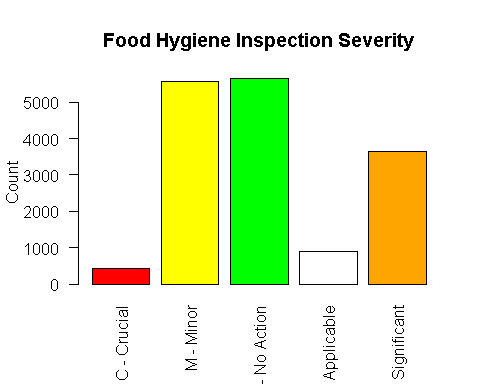
## CUISINE TYPE FREQUECY  
status <- table(Dinesafe$CUISINE\_TYPE)  
barplot(status, main="Cuisine Type", ylab="Frequency", beside=TRUE, col = rainbow(17),las=2, horiz=FALSE)



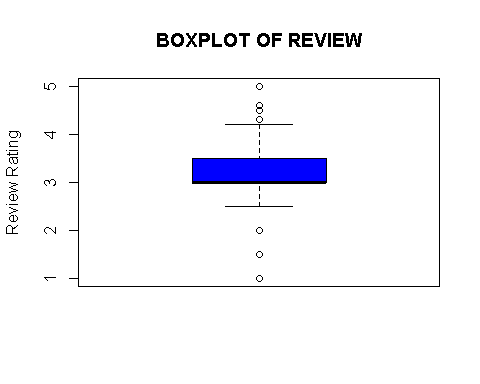
## Food Hygiene Inspection Result  
status <- table(Dinesafe$ESTABLISHMENT\_STATUS)  
barplot(status, main="Food Hygiene Inspection Result", xlab="Inspection Result", ylab="Establishment Count", col=c("red","yellow","green"), beside=TRUE)



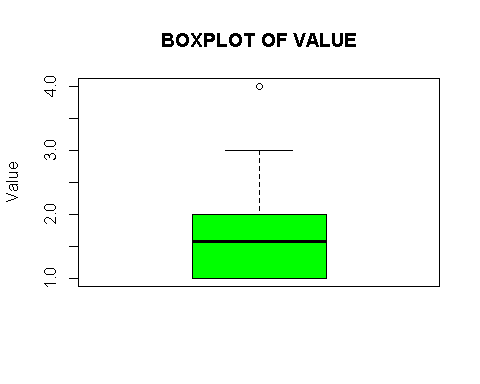
## Food Hygiene Inspection Result  
status1 <- table(Dinesafe$SEVERITY)  
barplot(status1, main="Food Hygiene Inspection Severity", xlab="", ylab="Count", col=c("red","yellow","green","White","Orange"), beside=TRUE,las=2)



boxplot(Dinesafe$REVIEW,  
 main = toupper("Boxplot of Review"),  
 ylab = "Review Rating",  
 col = "blue")



boxplot(Dinesafe$VALUE,  
 main = toupper("Boxplot of VALUE"),  
 ylab = "Value",  
 col = "green")



### Bivariate Analysis & Visualization

## Inspection outcome by Cuisine Stype  
Inspection\_Status\_By\_Cuisine\_Type <- table(Dinesafe$CUISINE\_TYPE, Dinesafe$ESTABLISHMENT\_STATUS)  
Inspection\_Status\_By\_Cuisine\_Type  
  
  
##Scatter plot of restaurant review rating vs value  
plot(Dinesafe$REVIEW, Dinesafe$VALUE, xlab="Review",ylab="Value",col=c("red","black"),cex=2)

