CKME136 - CAPSTONE - 1.2

Mohammed Amir

March 5, 2017

**Step 1. Data Load Process**

Dinesafe is an open dataset from City of Toronto Food Hygiene Inspection Report for the year 2015 and 2016. Address is a full address dataset extracted from google map for the Dinesafe food premises using google geocode. Load Dinesafe and Address Datasets, convert NULL values to NA.

Dinesafe = read.csv("D:/CAPSTONE/data/DineSafe\_02162017.csv", na.strings='NULL')  
Address = read.csv("D:/CAPSTONE/data/ADDRESS\_02262017.csv", na.strings='NULL')

**Step 2. Dataset Exploration Process**

**2.1 - Identify the column names for each datasets**

## List Column names  
colnames(Dinesafe)  
cat("\n")  
colnames(Address)

**2.2 - Identify database dimensions, Address has 7 columns and Dinesafe has 17 columns**

## Review dimension of dataset (Row by column)  
dim(Dinesafe)  
cat("\n")  
dim(Address)

**2.3 - Summarise the datasets.**

Find the min, max, median for quartile quantitative values, as well as identify word category counts for the categorical values.

## Review dataset summary  
summary(Dinesafe)  
cat("\n")  
summary(Address)

**2.4 - Identify the dataset structure such as int, factor, num for both datasets**

## Review dataset structure  
str(Dinesafe)  
cat("\n")  
str(Address)

**2.5 - Display top 5 sample data**

head(Dinesafe,5)  
head(Address,5)

**Step 3 - Merge Dinesafe and Address datasets based on establishment id column**

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

**3.1 - Analyise merged dataset**

dim(Dinesafe) ## Identify dimension  
cat("\n")  
str(Dinesafe) ## Identify structure  
cat("\n")   
table(Dinesafe$CUISINE\_TYPE, useNA = "always") ## Identify Cuisine Type  
cat("\n")   
table(Dinesafe$ESTABLISHMENT\_STATUS, useNA = "always") ## Identify Review Rating  
cat("\n")   
table(Dinesafe$DISTRICT, useNA = "always") ## Identify Districts  
cat("\n")   
table(Dinesafe$SEVERITY) ## Identify Severity Type

**Step 4 - Data Munging Step**

**4.1 - Data Cleaning and Transforming raw data into usable dataset**.

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

**4.2 - Change ACTION from factor to character to avoid error during imputation.**

## Convert Action column from factor to character type  
Dinesafe$ACTION = as.character(Dinesafe$ACTION)

**4.3 - Set catagorical level for Establishment status and Sevrity columns**

## Set Categorical Data Type Level for Establishment Status column  
Dinesafe$ESTABLISHMENT\_STATUS = factor(Dinesafe$ESTABLISHMENT\_STATUS,levels=c("Closed","Conditional Pass", "Pass"))  
cat("\n")  
  
## Set Categorical Data Type Level for Severity column  
Dinesafe$SEVERITY <- factor(Dinesafe$SEVERITY, levels = c("NA - Not Applicable", "N - No Action", "M - Minor", "S - Significant", "C - Crucial"))

**4.4 - Describe quantitative values in Reveiw and Rate columns**

library(Hmisc)

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, round.POSIXt, trunc.POSIXt, units

## Describe Review data  
describe(Dinesafe$REVIEW)  
cat("\n")  
  
## Describe Value data  
describe(Dinesafe$VALUE)  
cat("\n")

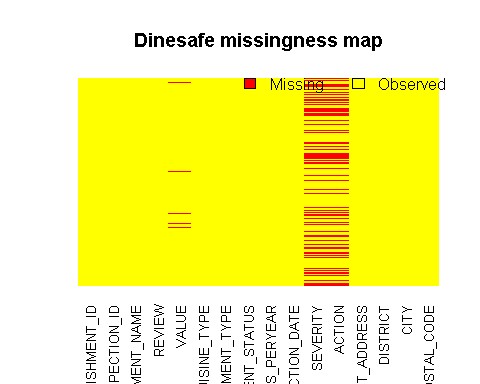
**4.5 - Identify missingness in Dinesafe dataset**

#Quantify missing values  
apply(Dinesafe, 2, function(x) sum(is.na(x)))  
cat("\n")  
  
# Plot missingness map using Amelia package  
library(Amelia)

## Loading required package: Rcpp

## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.7.4, built: 2015-12-05)  
## ## Copyright (C) 2005-2017 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

missmap(Dinesafe, col = c("Red","Yellow"), y.cex = 0.8, x.cex = 0.8, legend = TRUE, rank.order = "False" ,main = "Dinesafe missingness map", y.labels = NULL,y.at = NULL)



**4.6 - Show complete rows from dataset**

## Complete Case Rows with no missing (NA) value  
Complete\_Dinesafe <- Dinesafe[complete.cases(Dinesafe),]  
nrow(Complete\_Dinesafe)

**4.7 - Impute NA values in REVIEW column based on mean value of each cuisine type**

## 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)

**4.7 - Impute NA values in VALUE column based on mean value of each cuisine type**

## 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)

**4.8 - Impute missing severity and action columns where establishment status is PASS**

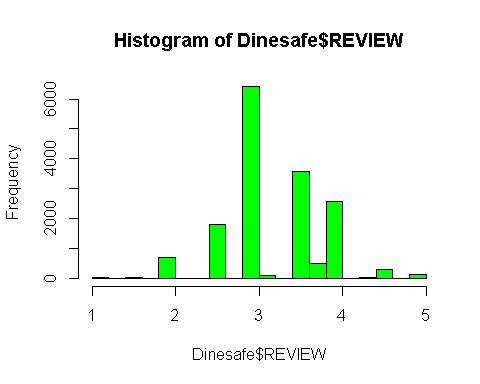
## Impute Severity column if it is NA and Establishment Status is PASS  
Dinesafe$SEVERITY[is.na(Dinesafe$SEVERITY) & Dinesafe$ESTABLISHMENT\_STATUS == "Pass"] = "NA - Not Applicable"  
cat("\n")  
## Impute Action column if it is NA and Establishment Status is PASS & Severity is No Action  
Dinesafe$ACTION[is.na(Dinesafe$ACTION) & Dinesafe$ESTABLISHMENT\_STATUS == "Pass" & Dinesafe$SEVERITY == "NA - Not Applicable"] = "No Action Required"

**4.9 - Check for incomplete rows**

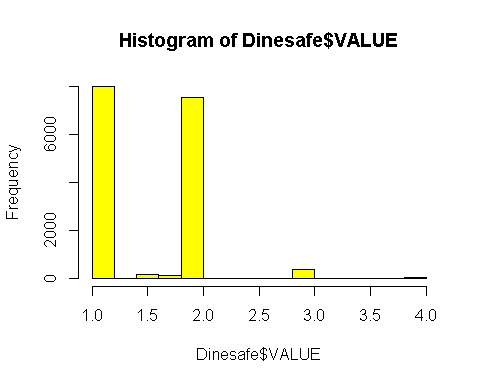
## Check for non complete case  
Dinesafe\_NA <- Dinesafe[!complete.cases(Dinesafe),]  
cat("\n")  
nrow(Dinesafe\_NA)

**Step 5 - Data Exploratory Analysis and Visualization 5.1 Univarient Data Analysis**

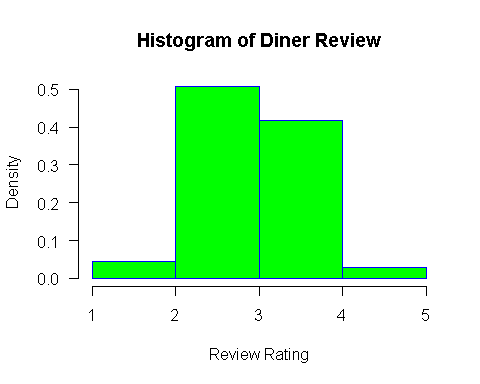
## The histogram graph of quantitative data in Dinesafe$Review shows that the data is normally distrubuted skewed to the left, where as Dinesafe$value shows that the data is not normally distrubuted.  
  
## Histogram graph  
hist(Dinesafe$REVIEW, col="GREEN")



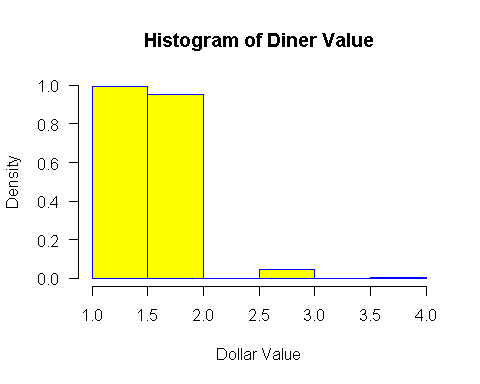
hist(Dinesafe$VALUE,col="YELLOW")



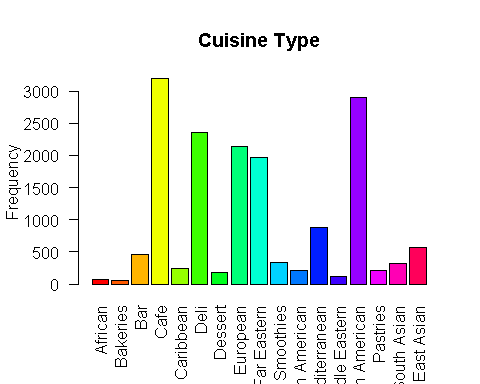
## Histogram Density graph  
hist(Dinesafe$REVIEW, xlab = "Review Rating", main="Histogram of Diner Review", border="blue", col="GREEN", las=1, breaks=5,prob = TRUE)



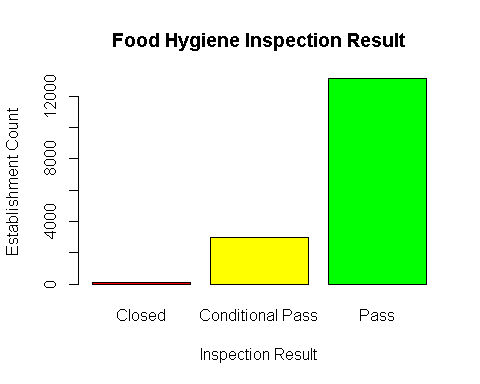
hist(Dinesafe$VALUE, xlab = "Dollar Value", main="Histogram of Diner Value", border="blue", col="YELLOW", las=1, breaks=5,prob = TRUE)



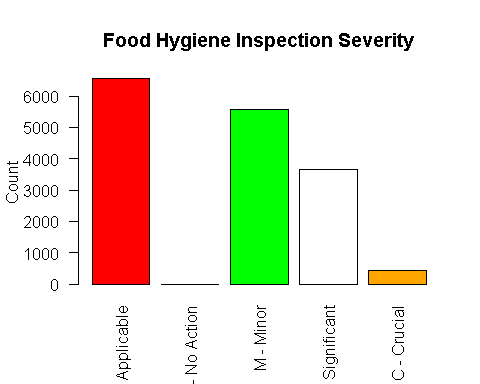
## Bar chart representation of a categorical values in Cusine Typek Inspection Result and Severity columns.  
  
## CUISINE TYPE FREQUECY  
Cuisin <- table(Dinesafe$CUISINE\_TYPE)  
barplot(Cuisin, main="Cuisine Type", ylab="Frequency", beside=TRUE, col = rainbow(17),las=2, horiz=FALSE)



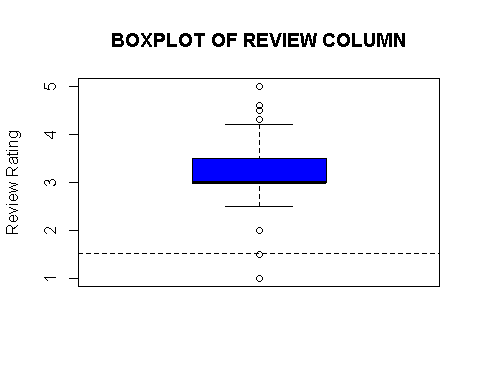
## Food Hygiene Inspection Result  
Inspection <- table(Dinesafe$ESTABLISHMENT\_STATUS)  
barplot(Inspection, main="Food Hygiene Inspection Result", xlab="Inspection Result", ylab="Establishment Count", col=c("red","yellow","green"), beside=TRUE)



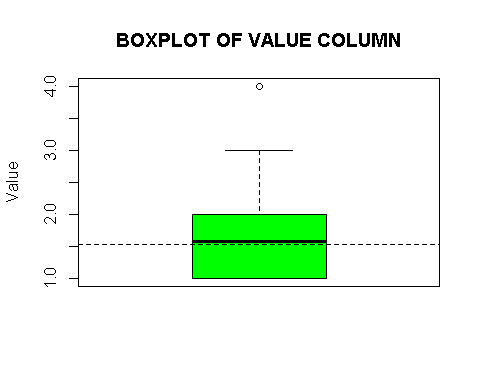
## Food Hygiene Inspection Severity  
Severity <- table(Dinesafe$SEVERITY)  
barplot(Severity, main="Food Hygiene Inspection Severity", xlab="", ylab="Count", col=c("red","yellow","green","White","Orange"), beside=TRUE,las=2)



## Boxplot of data distribution representation of Review and Value columns with its minimum, maximum, median, 1st quartile, 3rd quartile as well as outliers.  
  
## The Review boxplot shows that the rating value range is quite close, ie between 2.5 and 4 with most of the data is concertrated above the median value of 3. Outlier data are above 4 and below 2.5 and mean value is 1.5  
boxplot(Dinesafe$REVIEW,main = toupper("Boxplot of Review Column"),ylab = "Review Rating",col = "blue")  
abline(h=mean(Dinesafe$VALUE, na.rm = T), lty=2)



## The Value boxplot shows that it has one outlier and the data distriubtion is between 1 and 2 and the median and mean values are close to each other around 1.5  
boxplot(Dinesafe$VALUE,main = toupper("Boxplot of VALUE Column"),ylab = "Value",col = "green")  
abline(h=mean(Dinesafe$VALUE, na.rm = T), lty=2)



**5.2 Bivarient Data Analysis**

**5.2.1 - Mean & Standard deviation of Review and Value data against Estalishment status.**

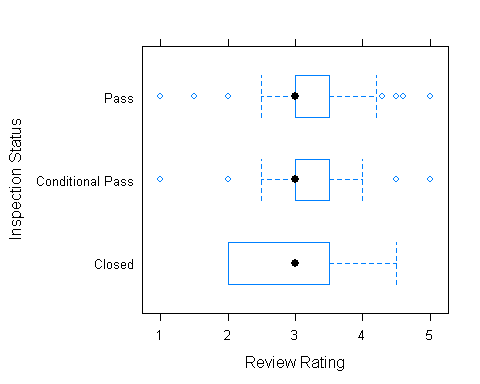
Based on the mean and standard deviation value of the review rating food premises that failed inspection had a mean review ratnig below the passed premises. Also failed food premises has a higher standard deviation value as compared to those who passed.

On the other hand the relationship between mean/standard deviation value and inspection outcome is not observer due to consistent result across all three values.

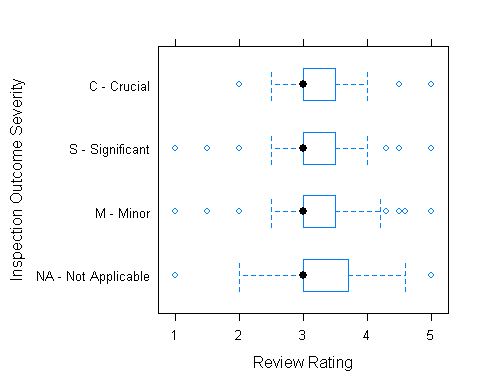
## Mean Review data against establishment inspection status  
tapply(Dinesafe$REVIEW , Dinesafe$ESTABLISHMENT\_STATUS, mean)  
cat("\n")  
## Standard Deviation of Review data against establishment inspection status  
tapply(Dinesafe$REVIEW , Dinesafe$ESTABLISHMENT\_STATUS, sd)   
cat("\n")  
## Mean value data against establishment inspection status  
tapply(Dinesafe$VALUE , Dinesafe$ESTABLISHMENT\_STATUS, mean)  
cat("\n")  
## Standard Deviation of Value data against establishment inspection status  
tapply(Dinesafe$VALUE , Dinesafe$ESTABLISHMENT\_STATUS, sd)

**5.2.2 - Categorical data analysis against the numerical Review Rating column using bwplot**

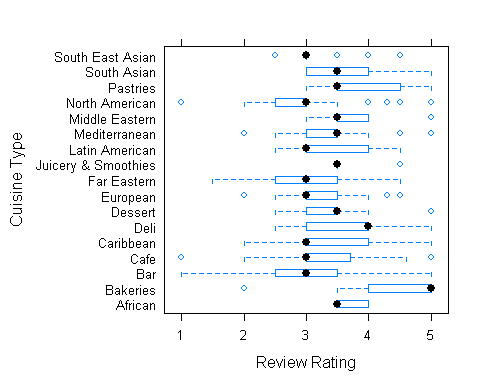
library(lattice)  
bwplot(ESTABLISHMENT\_STATUS ~ REVIEW, data = Dinesafe, ylab = "Inspection Status", xlab = "Review Rating")



bwplot(SEVERITY ~ REVIEW, data = Dinesafe, ylab = "Inspection Outcome Severity", xlab = "Review Rating")

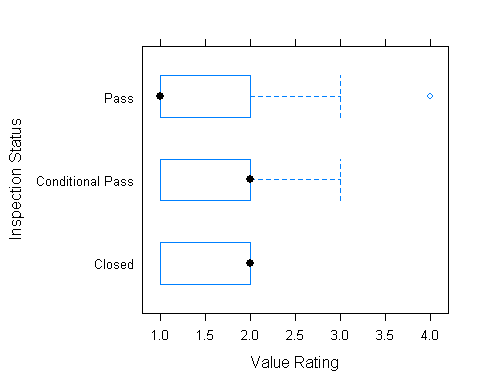


bwplot(CUISINE\_TYPE ~ REVIEW, data = Dinesafe, ylab = "Cuisine Type", xlab = "Review Rating")

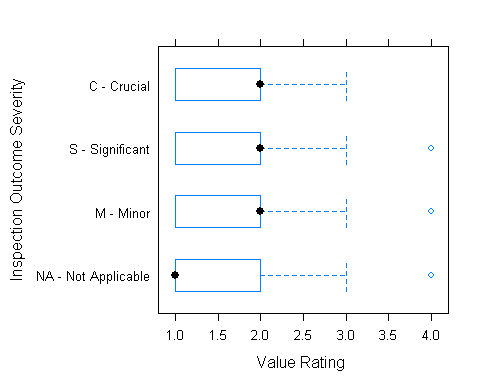


**5.2.3 - Categorical data analysis against the numerical Value Rating column using bwplot**

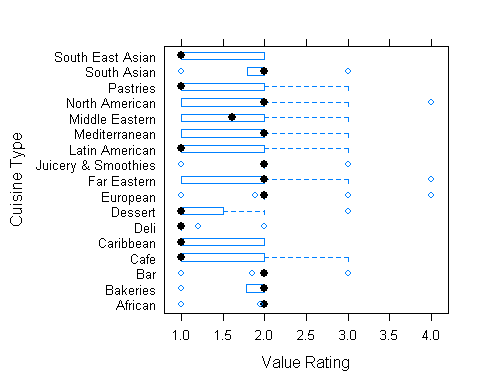
## Categorical data analysis  
bwplot(ESTABLISHMENT\_STATUS ~ VALUE, data = Dinesafe, ylab = "Inspection Status", xlab = "Value Rating")



bwplot(SEVERITY ~ VALUE, data = Dinesafe, ylab = "Inspection Outcome Severity", xlab = "Value Rating ")

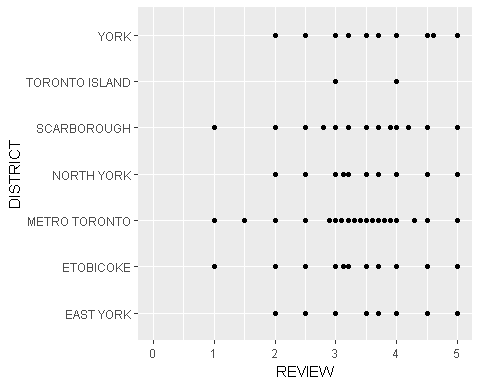


bwplot(CUISINE\_TYPE ~ VALUE, data = Dinesafe, ylab = "Cuisine Type", xlab = "Value Rating")

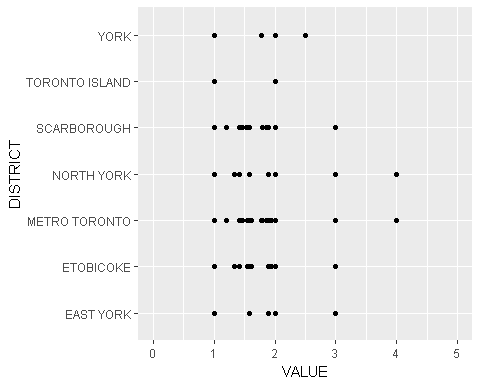


**5.2.4 - Categorical data analysis of District against the numerical data of Review and Value Rating column using qplot**

qplot(REVIEW, DISTRICT, data=Dinesafe) + xlim(0, 5)

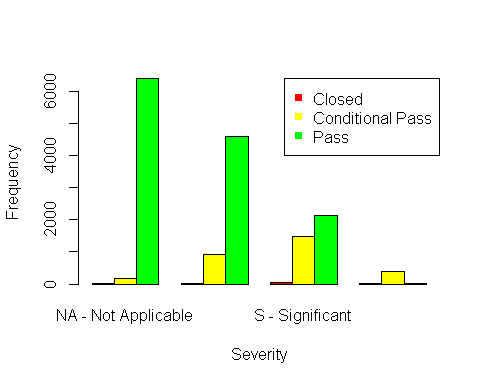


qplot(VALUE, DISTRICT, data=Dinesafe) + xlim(0, 5)



**5.2.5 - Relationship graphy between two categorical columns (INSPECTION STATUS AND SEVERITY)**

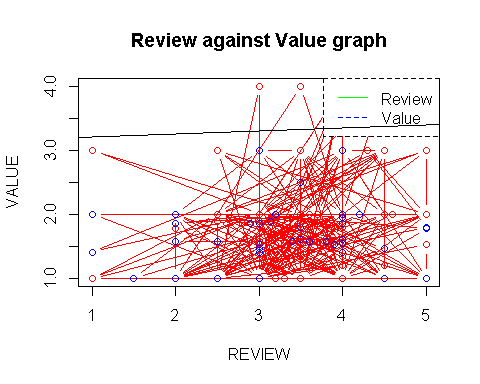
library(gmodels)  
joint = CrossTable(Dinesafe$ESTABLISHMENT\_STATUS, Dinesafe$SEVERITY)  
joint$t  
joint\_count = joint$t  
barplot(joint\_count, beside = TRUE, col = c("Red","Yellow","Green"),ylab = "Frequency", xlab = "Severity")  
legend("topright", c("Closed","Conditional Pass","Pass"), pch=15, col = c("Red","Yellow","Green"))



**5.2.6 - Scattered graph showing a relatship between two numerical values**

Based on the graph below there is no linear relationship between a restaurant review rating and value for money rating as the values are scattered all over the box and doesn't follow the simple linear regression model line.

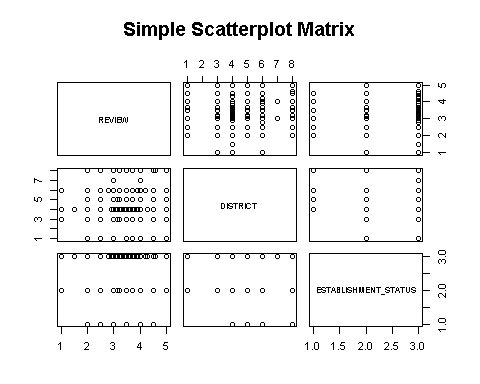
plot (Dinesafe$REVIEW, Dinesafe$VALUE, col = c("RED", "BLUE"), xlab="REVIEW", ylab="VALUE", main="Review against Value graph", type = "b")  
legend("topright", legend=c("Review", "Value"), col=c("green", "blue"), lty=1:2, cex=1, box.lty=2)  
abline(lm(Dinesafe$REVIEW ~ Dinesafe$VALUE))



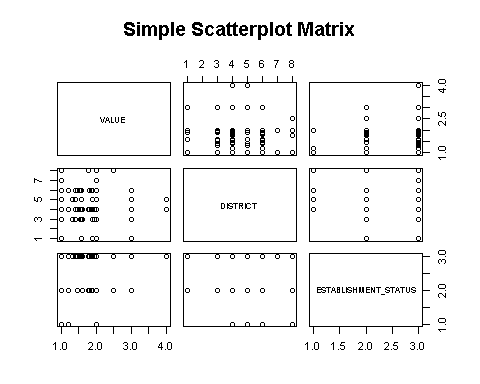
**5.3 - Multivariant Data Analysis**

**5.3.1 - Plot of a simple scattered matrix between three columns (REVIEW, DISTRICT & ESTABLISHMENT TYPE) & (VALUE, DISTRICT & ESTABLISHMENT TYPE)**

pairs(~REVIEW+DISTRICT+ESTABLISHMENT\_STATUS,data=Dinesafe, main="Simple Scatterplot Matrix")



pairs( ~VALUE+DISTRICT+ESTABLISHMENT\_STATUS,data=Dinesafe, main="Simple Scatterplot Matrix")



**5.3.2- Aggregation between multiple columns categorical and numerical values.**

head(aggregate(Dinesafe$REVIEW ~ Dinesafe$ESTABLISHMENT\_STATUS + Dinesafe$CUISINE\_TYPE + Dinesafe$DISTRICT, FUN=mean),10)  
head(aggregate(Dinesafe$REVIEW ~ Dinesafe$ESTABLISHMENT\_STATUS + Dinesafe$CUISINE\_TYPE, FUN=length),10)