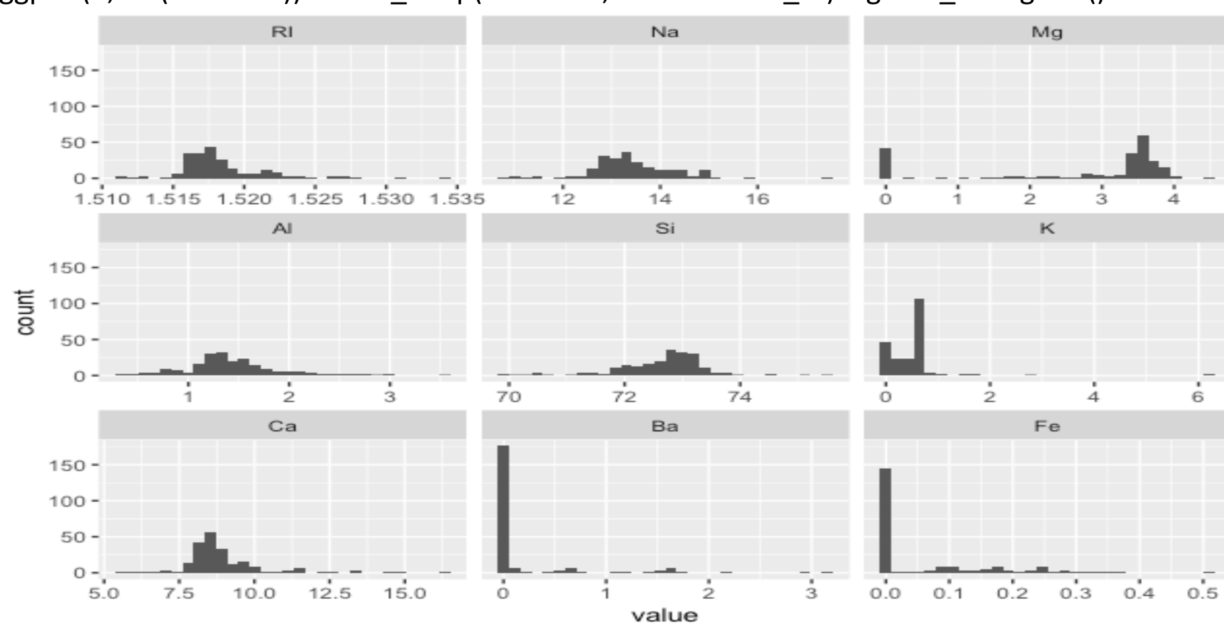
IDA Home Work 3
Pramod Aravind Byakod – 113436879

library(lattice)
library(car)
library(EnvStats)
library(corrplot)
library(ggbiplot)
library(mice)
library(VIM)
library(MASS)
library(Amelia)
library(ggplot2)
library(tidyr)
library(mlbench)
library(reshape2)

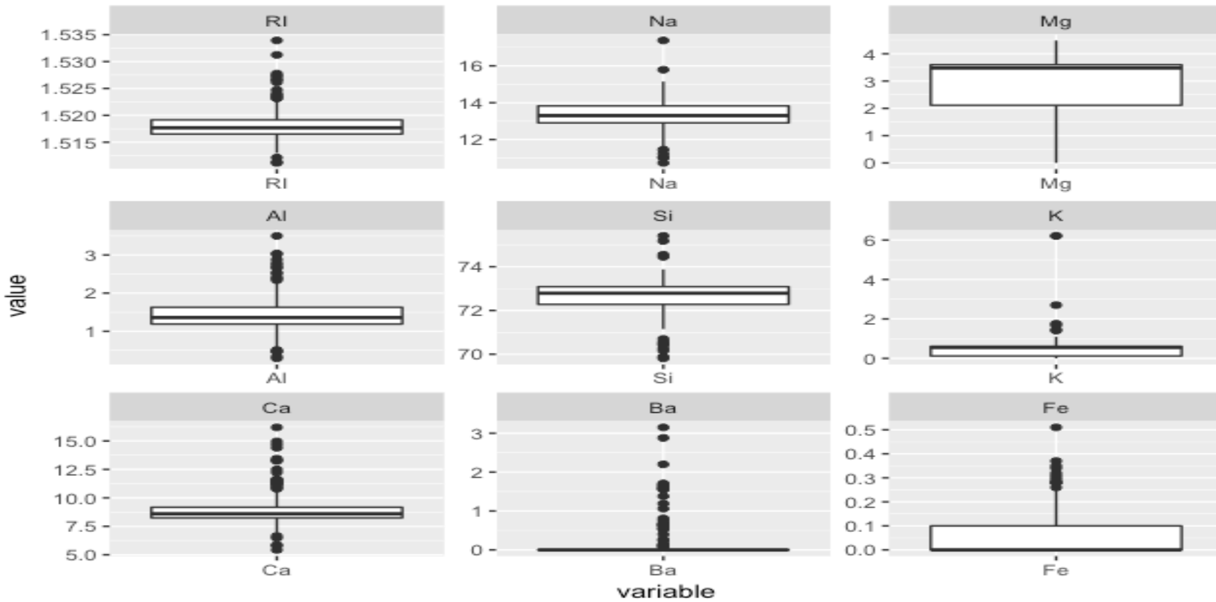
Question 1 – Glass Identification

1.a

data("Glass")
head(Glass)
```
#      RI    Na   Mg   Al    Si    K   Ca Ba   Fe
#1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00
#2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00
#3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00
#4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00
```
Glass = Glass[1:9]
d = melt(Glass)
ggplot(d,aes(x = value)) +facet_wrap(~variable,scales = "free_x") + geom_histogram()



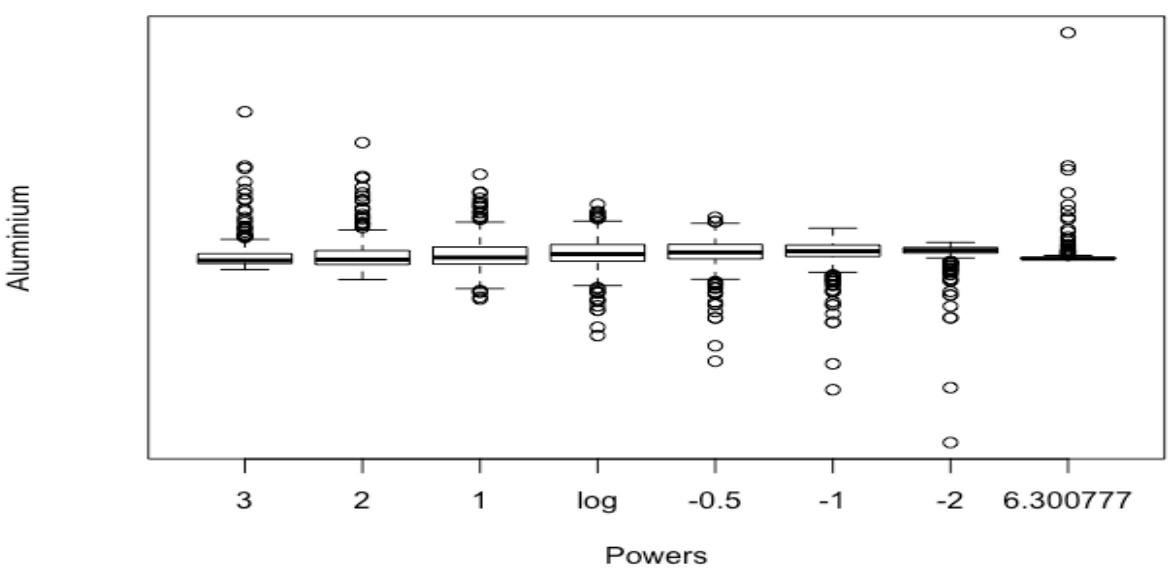
ggplot(d,aes(x = variable,y = value)) + facet_wrap(~variable, scale="free") + geom_boxplot()



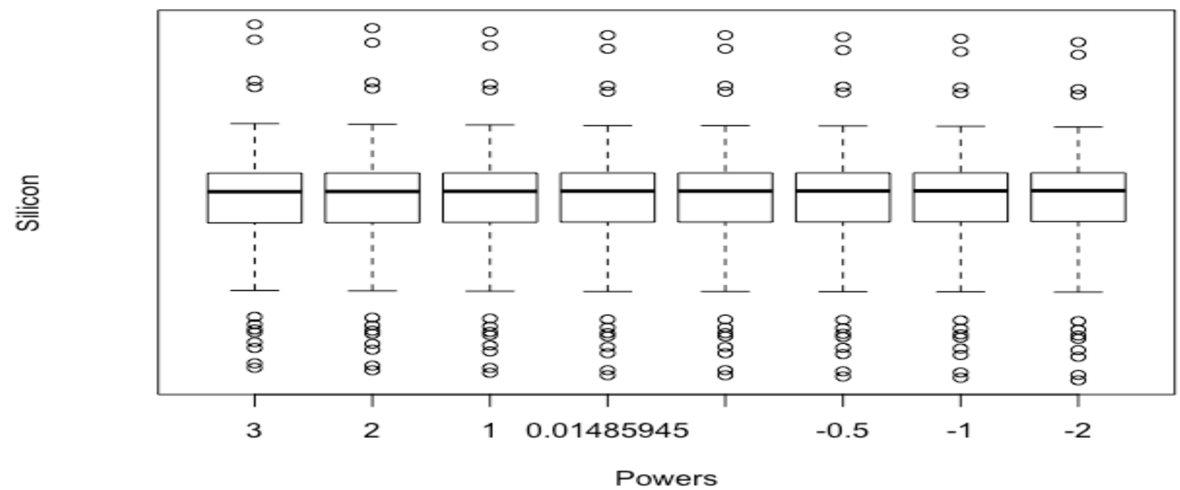
From the above plots RI, Na, Ca seems rightly skewed whereas Ba,Mg are left skewed. Few predictors follow normal distribution if we deal with outliers. Mg, Ba, Fe,K,Ca have many outliers compared to other predictors.

1.b.i

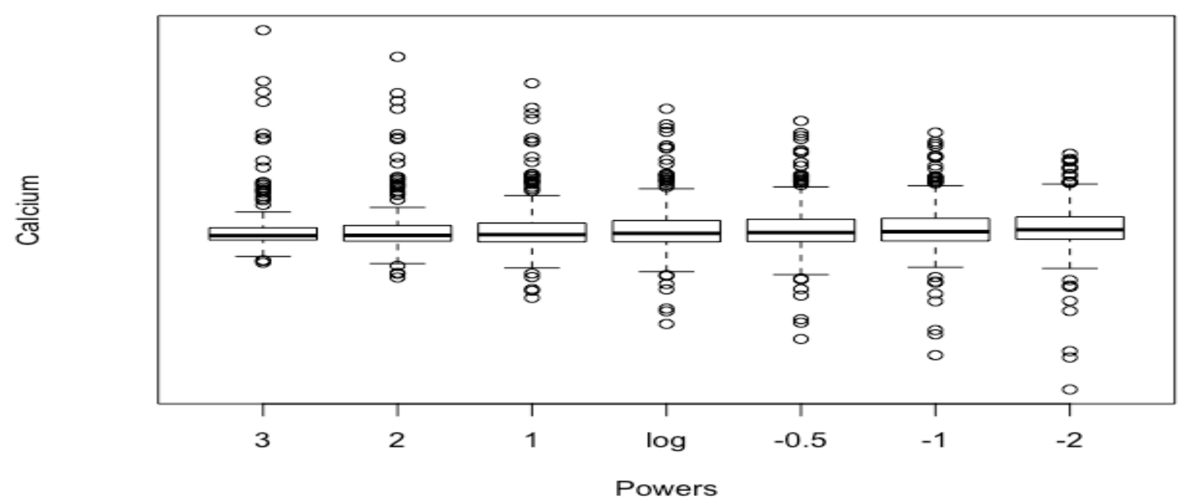
symbox(Glass$Al,data=Glass,powers=c(3,2,1,0,-0.5,-1,-2,6.300777),ylab="Aluminium")



symbox(Glass$Si,data=Glass,powers=c(3,2,1,0.01485945,0,-0.5,-1,-2),ylab="Silicon")



symbox(Glass$Ca,data=Glass,powers=c(3,2,1,0,-0.5,-1,-2),ylab="Calcium")



1.b.ii

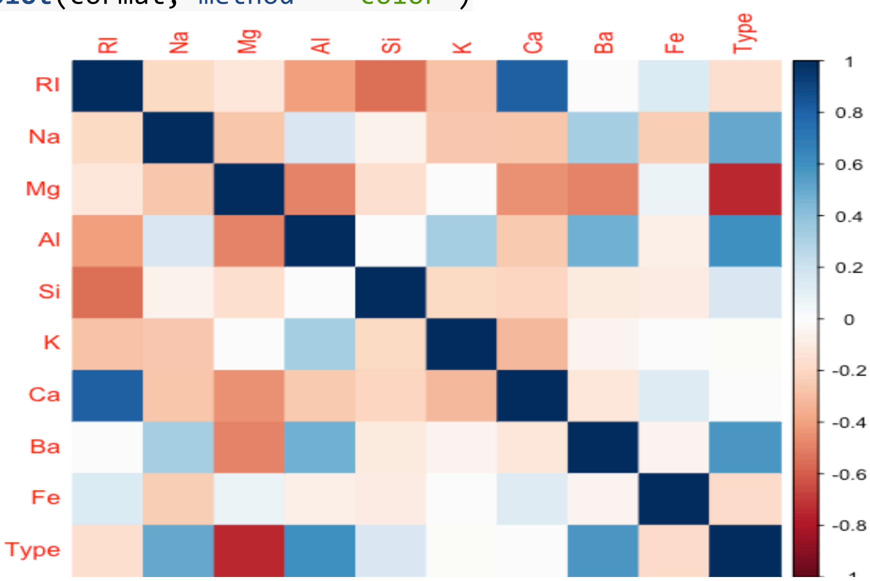
EnvStats::boxcox(Glass$Ca,optimize = TRUE, lambda=c(-5,7))
#$lambda $objective.name $objective
#[1] -0.8593591 [1] "PPCC" [1] 0.9390717

EnvStats::boxcox(Glass$Si,optimize = TRUE, lambda=c(-5,7))
#$lambda $objective.name $objective
#[1]7 [1] "PPCC" [1]0.9622907

EnvStats::boxcox(Glass$Al,optimize = TRUE, lambda=c(-5,7))
#$lambda $objective.name $objective
#1] 0.4844531 [1] "PPCC" [1] 0.9846485

1.c

```
cormat = cor(Glass)
corrplot(cormat, method = "color")
```



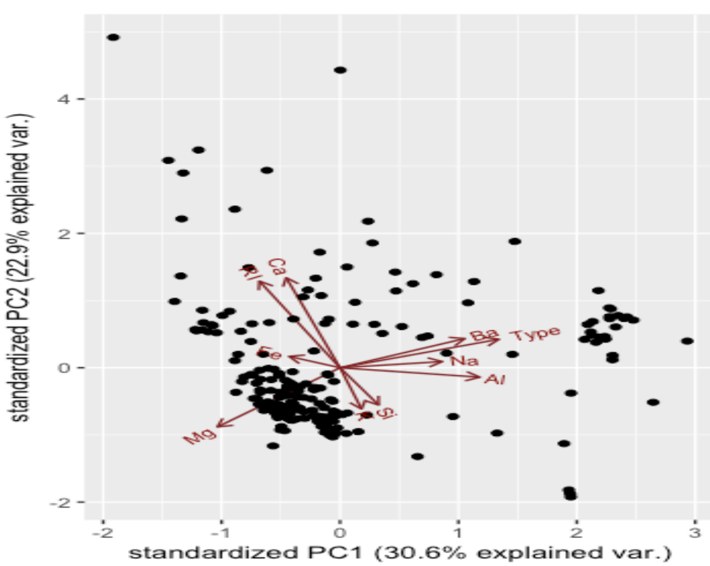
Correlation between Ca and RI is high as we can see from the plot.

eig_glass = eigen(cormat)
glasss_pca = prcomp(Glass, scale = T)
summary(glasss_pca)
#Importance of components:

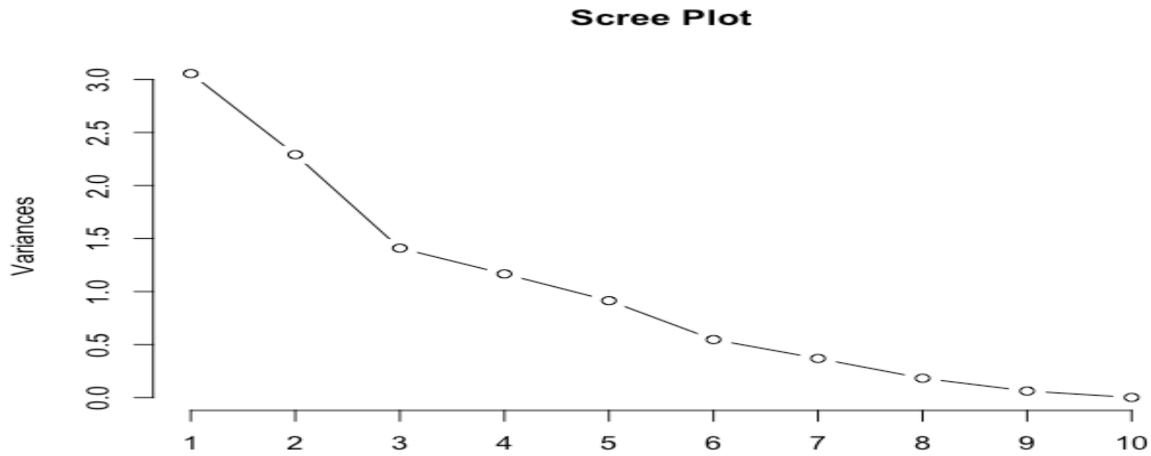
#	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
#Standard deviation	1.748	1.513	1.187	1.080	0.9560	0.7398	0.6078	0.4274	0.2491	0.0401
#Proportion of Variance	0.305	0.229	0.140	0.116	0.0914	0.0547	0.0369	0.0182	0.0062	0.0001
#Cumulative Proportion	0.305	0.534	0.675	0.792	0.8836	0.9384	0.9753	0.9936	0.9998	1.0000

94% of proportion of variance explained(PVE) by 4 pca components

ggbiplot(glasss_pca)



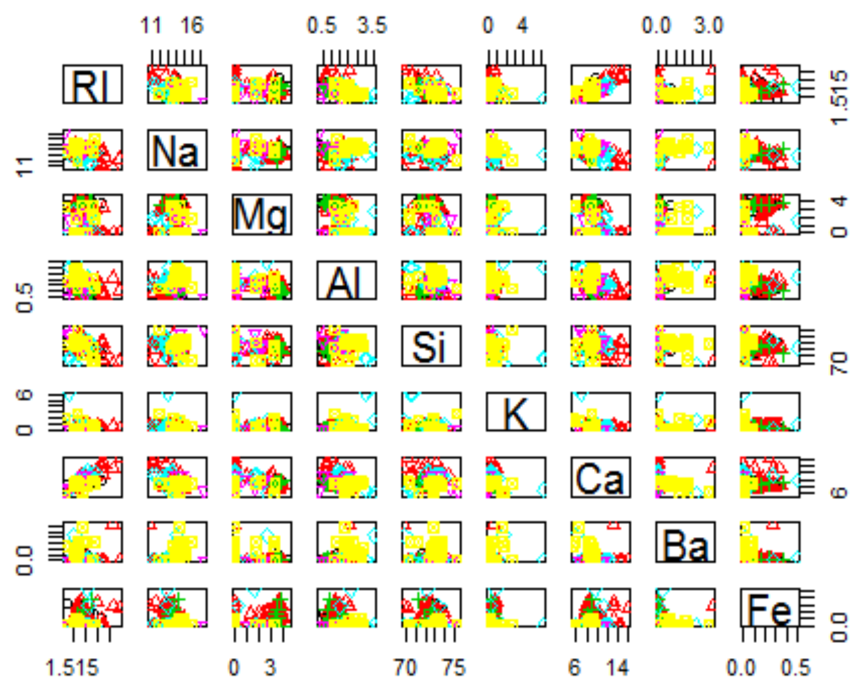
screeplot(glasss_pca, type = "line", npcs = 10, main = "Scree Plot")



From thje above two plots, Al,Na,Fe represents the PC1 component, where as PC2 represented by Ca,Ri,Si,K.

1.d

plot(Glass[,-10], col = Glass$Type, pch = Glass$Type)



glass.lda = lda(Type ~ RI+Na+Mg+Al+Si+K+Ca+Ba+Fe, data = Glass)
glass.lda.predict = predict(glass.lda, newdata = Glass[,-11])$class
glass.lda.predict

```
table(glass.lda.predict,Glass$Type)
#glass.lda.predict  1  2  3  5  6  7
#              1 52 17 11  0  1  1
#              2 15 54  6  5  2  2
#              3  3  0  0  0  0  0
#              5  0  3  0  7  0  1
#              6  0  2  0  0  6  0
#              7  0  0  0  1  0 25
glass.lda$counts
#1  2  3  5  6  7
#0 76 17 13  9 29
```

Confusion matrix shows that type1 is predicted 52 times truly positive, 54 times truly positive etc (diagonal elements of the matrix).
PCA and LDA both are dimensional reduction techniques. PCA is unsupervised and label agnostic, means that it treats the entire data as a whole, whereas LDA is supervised and tries to classify difference between classes.

Question 2 – Missing Data
2.a Regression using listwise deletion

```
dflistwise = freetrade
datawithoutmiss = na.omit(dflistwise)
outputa=lm(data=datawithoutmiss,tariff~year+country+polity+pop+gdp.pc+intresmi+signed+fiveop+usheg
summary(outputa)
#Multiple R-squared:  0.9311,	Adjusted R-squared:  0.9171
#F-statistic:  66.7 on 16 and 79 DF,  p-value: < 2.2e-16
```

2.b Regression using mean imputation

```
dfMean = freetrade
dfMean$tariff[which(is.na(dfMean$tariff))] = mean(dfMean$tariff,na.rm=T)
outputb=lm(data=dfMean,tariff~year+country+polity+pop+gdp.pc+intresmi+signed+fiveop+usheg)
summary(outputb)
#Multiple R-squared:  0.6412,	Adjusted R-squared:  0.5974
#F-statistic: 14.63 on 16 and 131 DF,  p-value: < 2.2e-16
```

2.c Regression using multiple imputation

```
dfMice = freetrade
imp = mice(dfMice,m=6,meth="mean",maxit = 10)
fitc = with(imp,lm(tariff~ year+country+polity+pop+gdp.pc+intresmi+signed+fiveop+usheg))
summary(fitc)
#Multiple R-squared:  0.6379,	Adjusted R-squared:  0.6002
#F-statistic: 16.95 on 16 and 154 DF, p-value: < 2.2e-16
```

2.d Comparison of the coefficients

summary(outputa) -> Multiple R-squared: 0.9311, Adjusted R-squared: 0.9171
summary(outputb) -> Multiple R-squared: 0.6412, Adjusted R-squared: 0.5974
summary(fitc) -> Multiple R-squared: 0.6379, Adjusted R-squared: 0.6002
Comparison shows that listwise deletion has high R-squared value where as multiple imputation has less value.

Question 4 - Kaggle.com

4.a Data Selection

Data Selected - Crime Data for Philadelphia, it is extracted from Kaggle

URL - www.kaggle.com/mchirico/philadelphiacrimedata

Data name – crime.csv

This data set includes the data related to crimes committed in Philadelphia. Includes details of crime date, crime time, place where crime committed, police district, location blocked etc.

4.b Explore Data

```
summary(crime_data)
 #Dc_Dist          Psa                Dispatch_Date_Time  Dispatch_Date      Dispatch_Time
 #Min.   : 1.00    Length:2237605     Length:2237605      Length:2237605     Length:2237605
 #1st Qu.: 9.00    Class :character   Class :character    Class :character   Class :character
 #Median :16.00    Mode  :character   Mode  :character    Mode  :character   Mode  :character
 #Mean   :17.27
 #3rd Qu.:24.00
 #Max.   :92.00
 #Hour             Dc_Key             Location_Block      UCR_General    Text_General_Code
 #Min.   : 0.00    Min.   :1.998e+11  Length:2237605      Min.   : 100   Length:2237605
 #1st Qu.: 9.00    1st Qu.:2.008e+11  Class :character    1st Qu.: 600   Class :character
 #Median :14.00    Median :2.011e+11  Mode  :character    Median : 800   Mode  :character
 #Mean   :13.16    Mean   :2.011e+11                      Mean   :1271
 #3rd Qu.:19.00    3rd Qu.:2.014e+11                      3rd Qu.:1800
 #Max.   :23.00    Max.   :2.018e+11                      Max.   :2600
                                                         NA's   :663
 #Police_Districts  Month              Lon               Lat
 #Min.   : 1.00     Length:2237605     Min.   :-75.28    Min.   :39.87
 #1st Qu.: 8.00     Class :character   1st Qu.:-75.19    1st Qu.:39.96
 #Median :12.00     Mode  :character   Median :-75.16    Median :39.99
 #Mean   :12.06                        Mean   :-75.15    Mean   :39.99
 #3rd Qu.:17.00                        3rd Qu.:-75.12    3rd Qu.:40.03
 #Max.   :22.00                        Max.   :-74.96    Max.   :40.14
 #NA's   :19930                        NA's   :17349     NA's   :17349
```

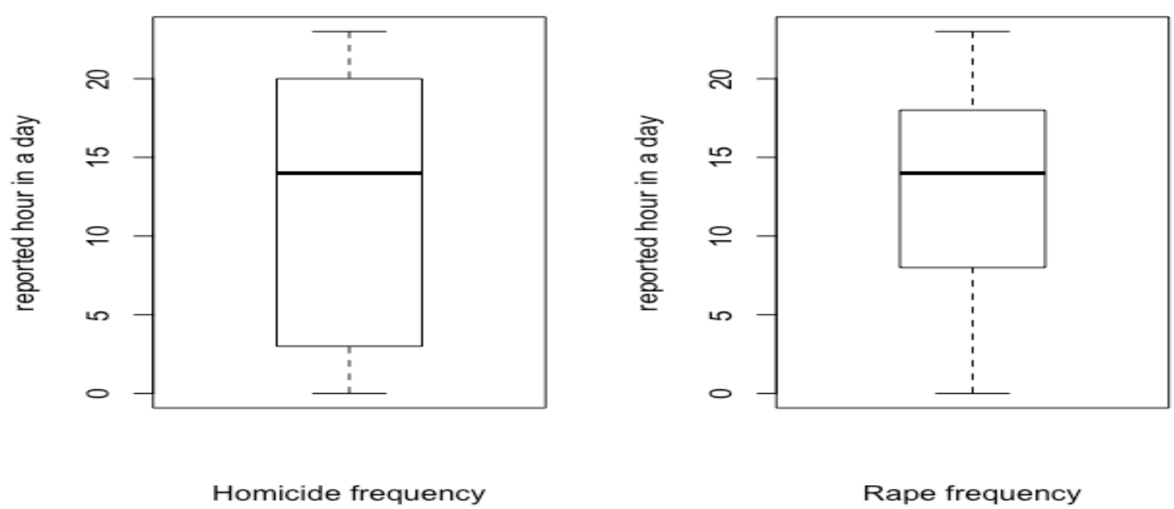
Below are the columns present in dataset crime_data

```
colnames(crime_data)
#[1] "Dc_Dist"          "Psa"              "Dispatch_Date_Time" "Dispatch_Date"
#[5] "Dispatch_Time"    "Hour"             "Dc_Key"             "Location_Block"
#[9] "UCR_General"      "Text_General_Code"  "Police_Districts"   "Month"
#[13] "Lon"             "Lat"
```

Number of columns in crime_data
ncol(crime_data)
# [1] 14
Number of rows in crime_data
nrow(crime_data)
# [1] 2237605

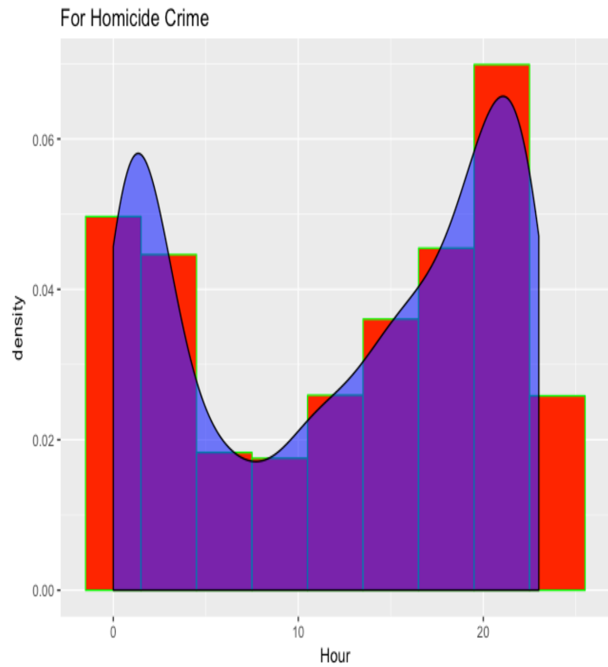
Now let's extract the "Rape" and "Homicide" crime data into separate data frames.
crime_homicide = crime_data[grepl("Homicide", crime_data$Text_General_Code, ignore.case = T),]
crime_rape = crime_data[crime_data$Text_General_Code == "Rape",]

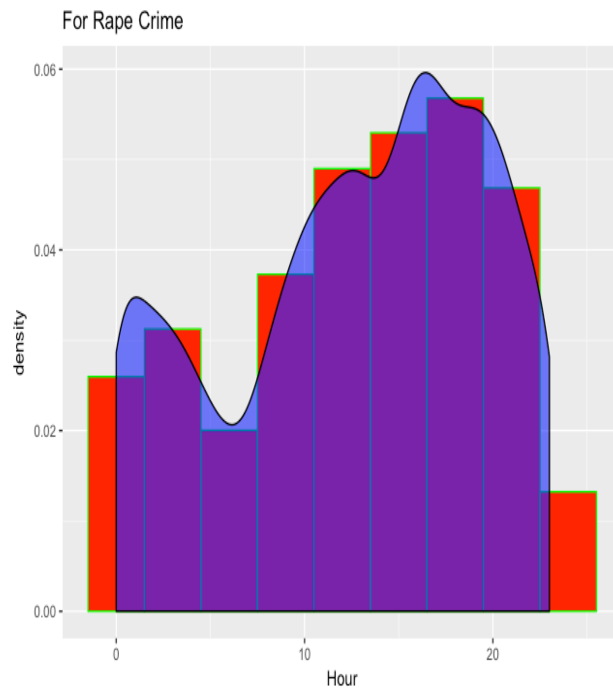
Let's boxplot the frequency of both the crimes happening in hours of a day. For that we load a function called "boxplot.with.outlier.label" from github.
source("https://raw.githubusercontent.com/talgalili/R-code-snippets/master/boxplot.with.outlier.label.r")
boxplot.with.outlier.label(crime_homicide$Hour,ylab = "reported hour in a day", xlab = "Homicide frequency")
boxplot.with.outlier.label(crime_rape$Hour,ylab = "reported hour in a day", xlab = "Rape frequency")



We could clearly observe in the plots that there are no outliers detected.

To visualize the data we will make use of ggplot
ggplot(crime_homicide, aes(x=Hour)) + geom_histogram(aes(y=..density..),binwidth= 3,colour="green", fill="red") +geom_density(alpha=0.6, fill="blue")
ggplot(crime_rape, aes(x=Hour)) + geom_histogram(aes(y=..density..),binwidth= 3,colour="green", fill="red") +geom_density(alpha=0.6, fill="blue")





Information on missing value counts

```
apply(crime_data,2, function(x){table(is.na(x))})
$Dc_Dist   $Psa     $Dispatch_Date_Time  $Dispatch_Date  $Dispatch_Time  $Hour
  FALSE     FALSE     FALSE                FALSE           FALSE           FALSE
2237605   2237605  2237605               2237605         2237605         2237605

$Dc_Key   $Location_Block  $UCR_General      $Text_General_Code  $Police_Districts
  FALSE     FALSE            FALSE    TRUE    FALSE                FALSE     TRUE
2237605   2237605          2236942   663    2237605              2217675   19930

$Month    $Lon                $Lat
  FALSE     FALSE     TRUE      FALSE     TRUE
2237605   2220256  17349    2220256  17349
```

In the above result TRUE means missing value, FALSE means data is available

Question 3 House prices data

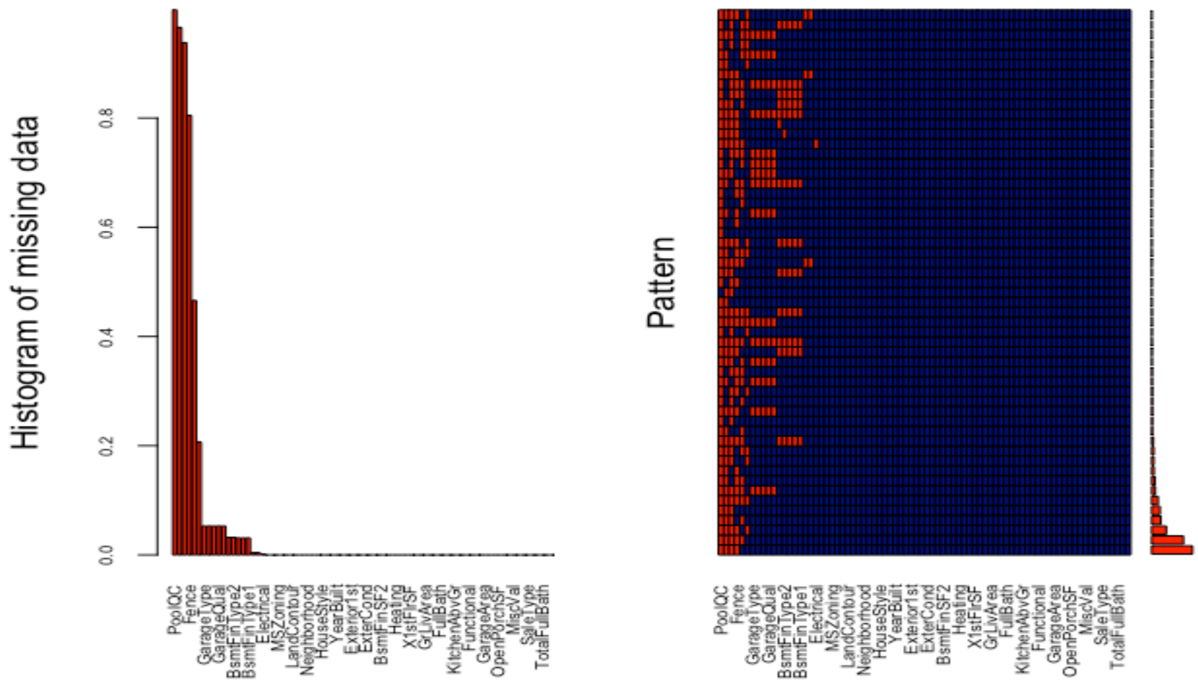
3.a

housingData = read.csv("housingData.csv", header = TRUE, sep = ",")
summary(housingData)

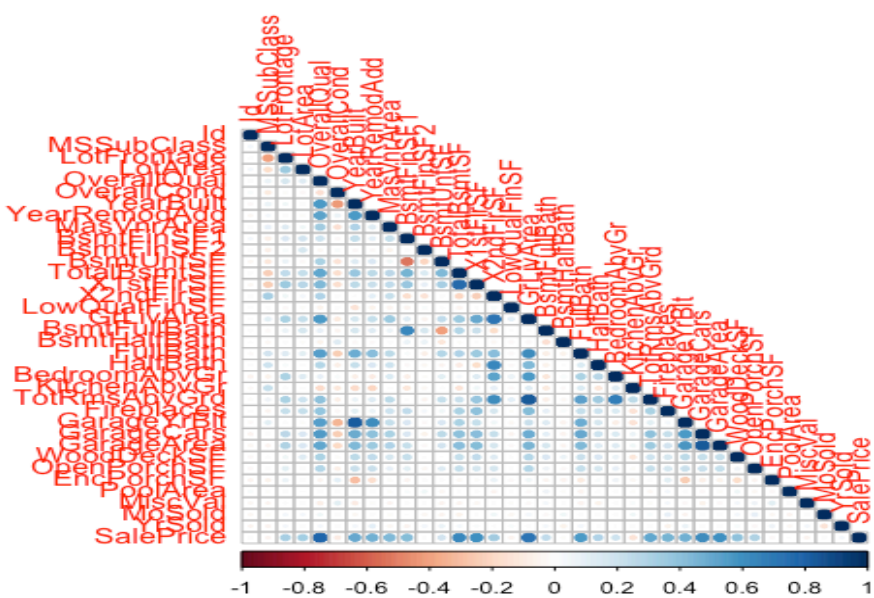
From the summary of the data we can say that few columns have missing values and those missing values can be categorized among **categorical and numerical values**

Let's have a closer look at missing data.

aggr(housingData, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(housingData), cex.axis=.6, gap=4, ylab=c("Histogram of missing data","Pattern"))

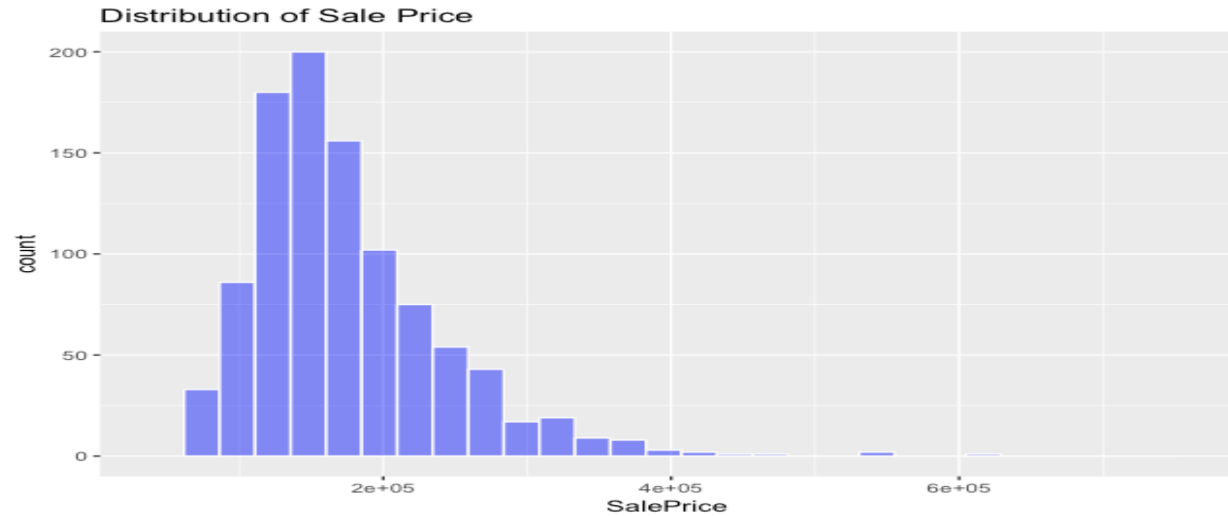


The missing values indicate that majority of the houses do not have alley access, no pool, no fence and no elevator, 2nd garage, shed or tennis court that is covered by the MiscFeature.
To enhance the visualization, will form the correlation matrix and corrplot to plot the details. Let's plot only plot the correlation between the numeric variables
housing_cor_numerics = cor(na.omit(housingData[,numerical_var]))
corrplot:: corrplot(housing_cor_numerics, method="circle", type="lower", insig = "blank")



Consider sales price as the primary attribute, let's explore and visualize more on it.

ggplot(data=housingData, aes(x=SalePrice)) + geom_histogram(color='white', alpha=0.5, fill='blue') +scale_x_continuous(limits = c(min(housingData$SalePrice),max(housingData$SalePrice))) +ggtitle('Distribution of Sale Price')

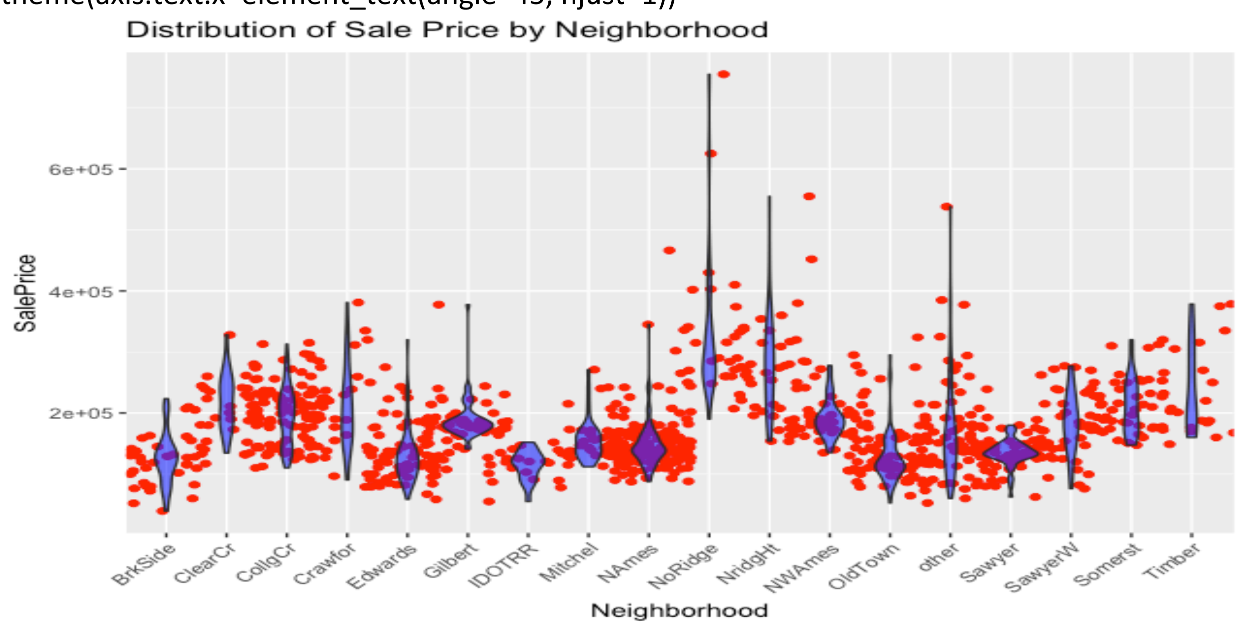


As we look at the plot of sales price, we can say that Sales Price is right skewed.

When people consider buying homes, usually the location has been constrained to a certain area such as not too far from the workplace. I would consider this variable also strong feature.

Distribution of prices how the price range is changing by neighborhood wise. This plot gives the clear picture of the max and min sales price distribution by grouping neighbors

ggplot(housingData, aes(x=Neighborhood, y=SalePrice)) +geom_jitter(color='red', width=0.7) +geom_violin(fill='blue', alpha=0.6) +ggtitle('Distribution of Sale Price by Neighborhood') +scale_y_continuous(limits = c(min(housingData$SalePrice),max(housingData$SalePrice))) + theme(axis.text.x=element_text(angle=45, hjust=1))



3.b

Now let’s create few new features

Sale price depends on the number of years house used, rather than Yearbuilt and YearSold.

housingData$YearsUsed = housingData$YearBuilt - housingData$YrSold

Gives total number of fullBath rooms

housingData$TotalFullBath = housingData$BsmtFullBath - housingData$FullBath

Gives total Floor area.

housingData$Floorsqft = housingData$X1stFlrSF + housingData$X2ndFlrSF

Summing overall qual & OverallCond

housingData$overall = housingData$OverallQual + housingData$OverallCond

3.c

Number of years the house used is one of the important factor for the buyers. Though we have year built and year sold data, having YearsUsed attribute makes the buyers life easier.

People also consider full bath rooms as one of the main factors while buying, so we add TotalFullBath attribute to the data frame.

Floorsqrft gives total area of the floors in square feet including 1^{st} and 2^{nd} floors.

We add rating for material and overall condition of the house to form another rating which gives overall rating of the house