Homework #4

Azadeh Glanpour

September 25, 2018

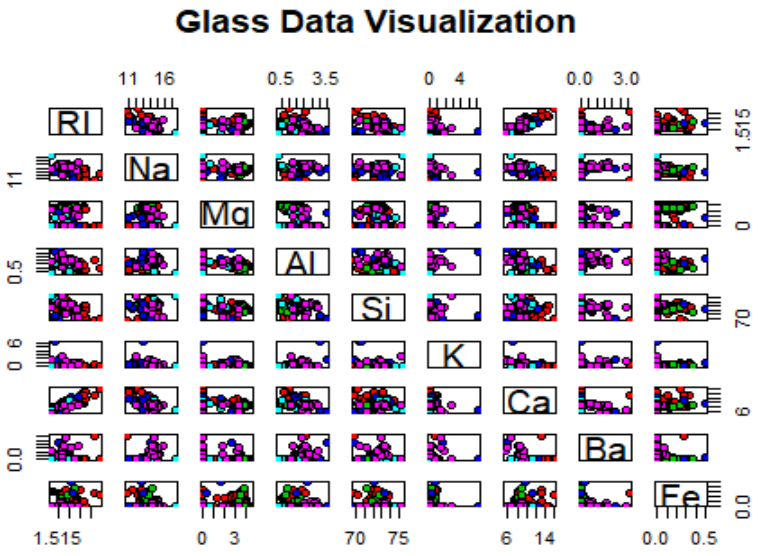
Problem 1

1a)

```
# Load data from the package "mlbench"
data(Glass)
```

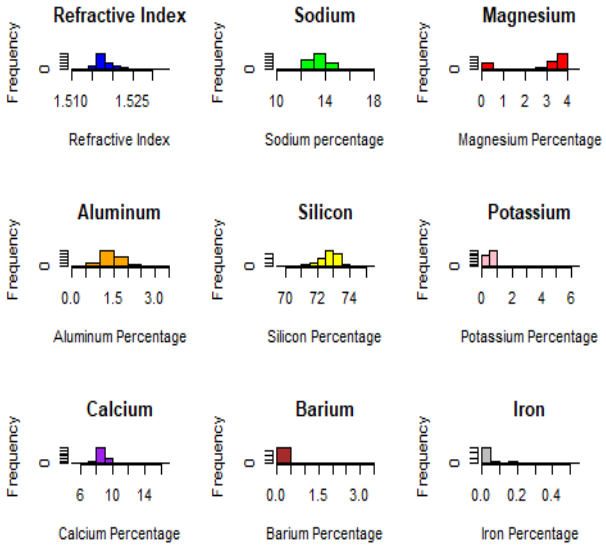
Some of ions like *Mg*, *Ba*, and *Fe* have values of zero. Looking into chemistry of the glasses releavs that these ions are not essential part of the glasses, but they carries some identifications such as color of galss. Therefore; they would be very useful to classify the glasses as we have a variable as type.

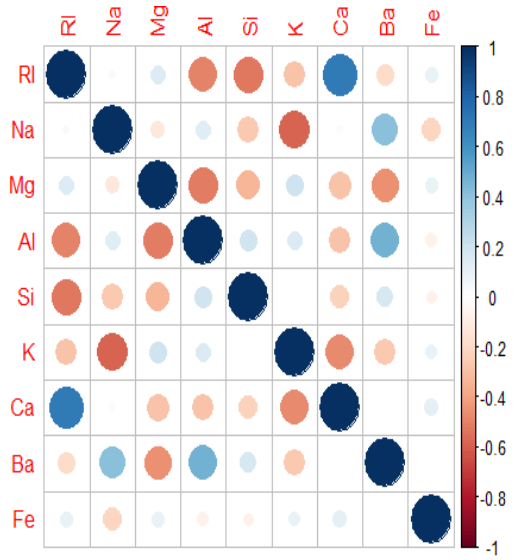
```
pairs(Glass[1:9], main = "Glass Data Visualization", pch=21, bg = Glass$Type
)
```




```
# Histogram - separated
par(mfrow=c(3,3))
hist(Glass$RI, main="Refractive Index", xlab="Refractive Index",col="blue")
hist(Glass$Na, main="Sodium", xlab="Sodium percentage",col="green")
hist(Glass$Mg, main="Magnesium", xlab="Magnesium Percentage",col="red")
hist(Glass$Al, main="Aluminum", xlab="Aluminum Percentage",col="orange")
hist(Glass$Si, main="Silicon", xlab="Silicon Percentage",col="yellow")
hist(Glass$K,  main="Potassium", xlab="Potassium Percentage",col="pink")
```

```
hist(Glass$Ca, main="Calcium", xlab="Calcium Percentage",col = "purple")
hist(Glass$Ba, main="Barium", xlab="Barium Percentage",col="brown")
hist(Glass$Fe, main="Iron", xlab="Iron Percentage",col = "gray")
```

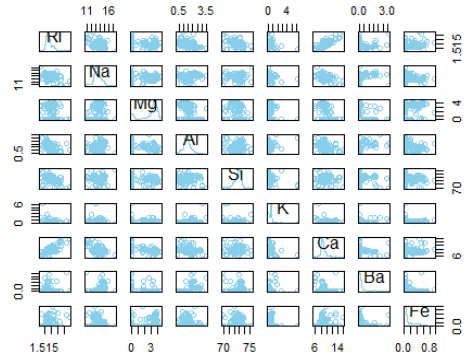


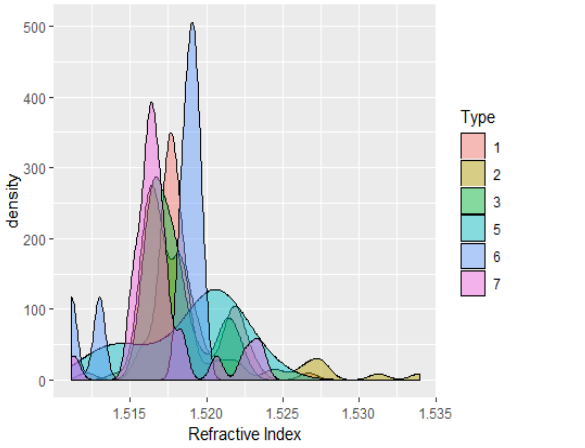


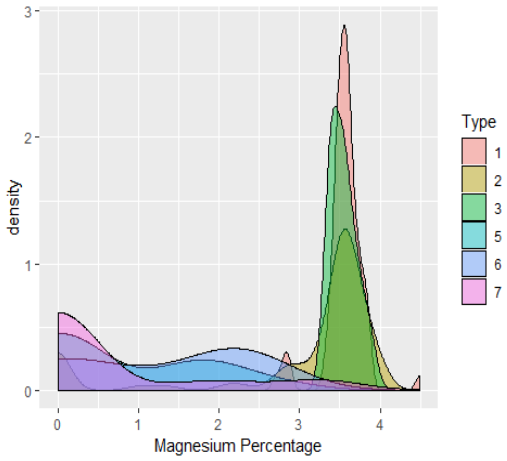

```
# Check the correlations
correalation<-cor(Glass[,1:9],method = "kendall")
correalation<-cor(Glass[,1:9],method = "pearson")
correalation<-cor(Glass[,1:9],method = "spearman")
corrplot(correalation)

# Breif view for binary scatter
scattmatrixMiss(Glass[,1:9])
```

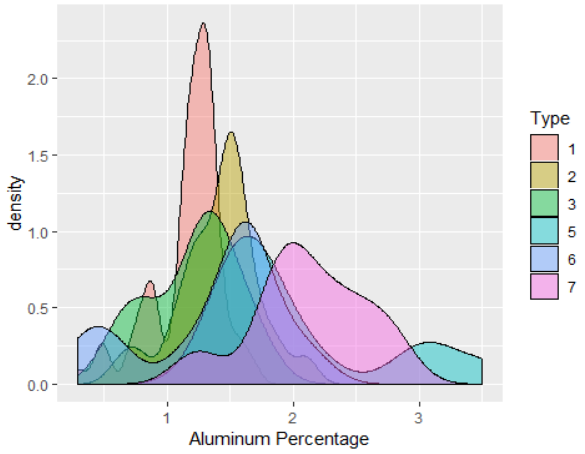
```
Glass.melt<- melt(Glass)

par(mfrow=c(3,3))
ggplot(Glass.melt[Glass.melt$variable=="RI",], aes(x=value, fill=Type)) +
geom_density(alpha=0.45) + labs(x="Refractive Index")

ggplot(Glass.melt[Glass.melt$variable=="Na",], aes(x=value, fill=Type)) +
geom_density(alpha=0.45) + labs(x="Sodium Percentage")

ggplot(Glass.melt[Glass.melt$variable=="Mg",], aes(x=value, fill=Type)) +
geom_density(alpha=0.45) + labs(x="Magnesium Percentage")

ggplot(Glass.melt[Glass.melt$variable=="Al",], aes(x=value, fill=Type)) +
geom_density(alpha=0.45) + labs(x="Aluminum Percentage")

ggplot(Glass.melt[Glass.melt$variable=="Si",], aes(x=value, fill=Type)) +
geom_density(alpha=0.45) + labs(x="Silicon Percentage")
```

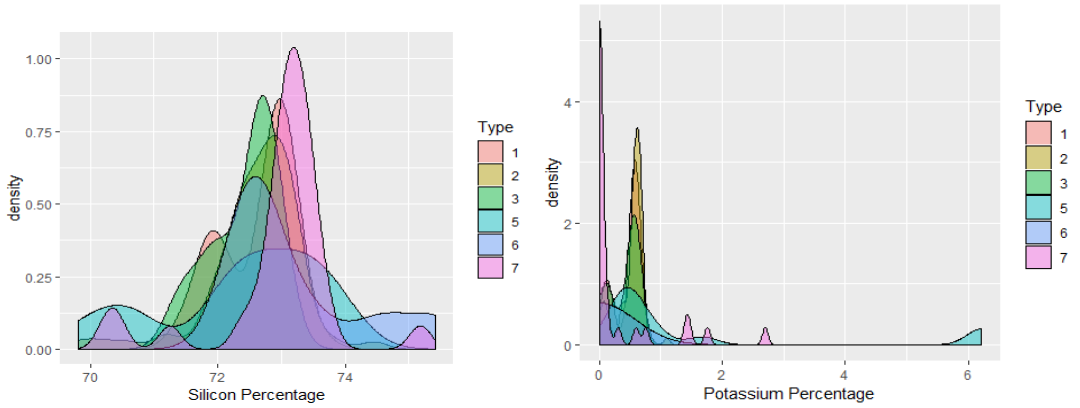
```r
ggplot(Glass.melt[Glass.melt$variable=="K", ], aes(x=value, fill=Type)) +
geom_density(alpha=0.45) + labs(x="Potassium Percentage")

ggplot(Glass.melt[Glass.melt$variable=="Ca",], aes(x=value, fill=Type)) +
geom_density(alpha=0.45) + labs(x="Calcium Percentage")

ggplot(Glass.melt[Glass.melt$variable=="Ba",], aes(x=value, fill=Type)) +
geom_density(alpha=0.45) + labs(x="Barium Percentage")

ggplot(Glass.melt[Glass.melt$variable=="Fe",], aes(x=value, fill=Type)) +
geom_density(alpha=0.45) + labs(x="Iron Percentage")
```

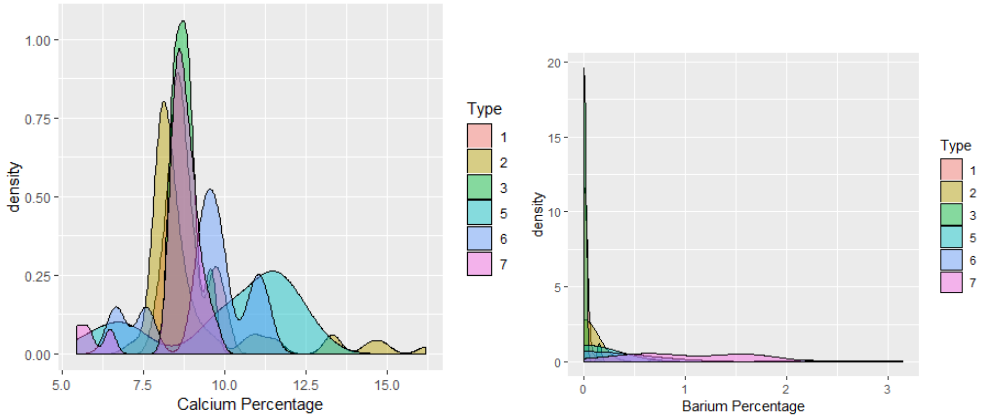


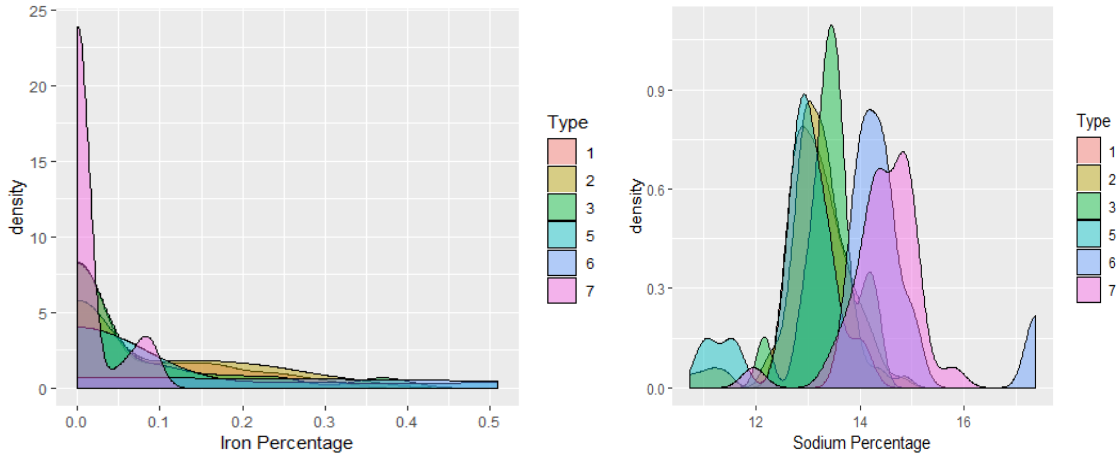






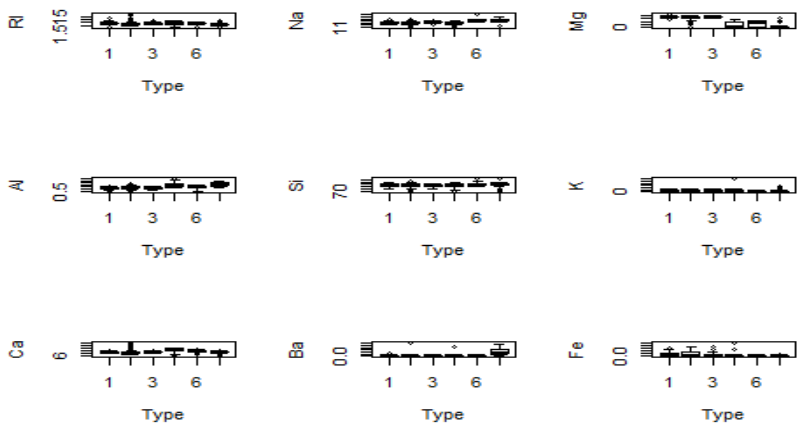




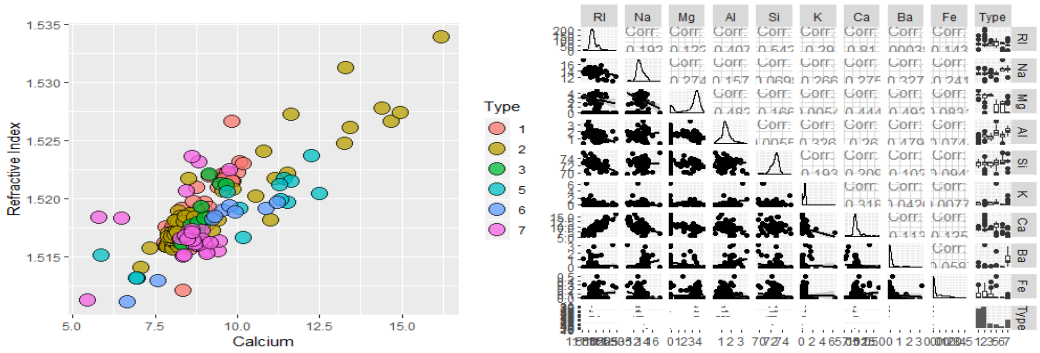



```r
# Boxplots
par(mfrow=c(3,3))                    # Divides the screen into three sections
boxplot(data=Glass, RI ~ Type, xlab = "Type", ylab = "RI")
boxplot(data=Glass, Na ~ Type, xlab = "Type", ylab = "Na")
boxplot(data=Glass, Mg ~ Type, xlab = "Type", ylab = "Mg")
boxplot(data=Glass, Al ~ Type, xlab = "Type", ylab = "Al")
boxplot(data=Glass, Si ~ Type, xlab = "Type", ylab = "Si")
boxplot(data=Glass, K ~ Type, xlab = "Type", ylab = "K")
boxplot(data=Glass, Ca ~ Type, xlab = "Type", ylab = "Ca")
```

```
boxplot(data=Glass, Ba ~ Type, xlab = "Type", ylab = "Ba")
boxplot(data=Glass, Fe ~ Type, xlab = "Type", ylab = "Fe")
```




```
ggplot(data=Glass, aes(x=Ca,y=RI))+ geom_point(aes(fill=Type), alpha=I(.75),
colour="black", pch=21, size=5)+
labs(x="Calcium",y="Refractive Index")

ggpairs(Glass, lower=list(continuous="smooth"),axisLabels='show')
```

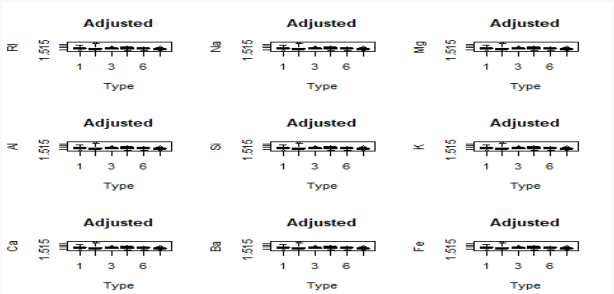


Some tecniques were applied to investigate the distribution, and the relationship between the predictor variables. The following picture shows that there are not exist strong linear relationship between the predictors except ***Refractive Index*** and ***Calcium***. This implies that "calcium" and "reflactive index" are correlated and we may use one of them or combination of them as a single predictor which one is easier.

The weak linear relationship between predictors never deny existance of nonlinear correlations. Visualization helps us to realize whether any pattern is available in plotted data. By looking at the plots, I would say some nonlinear correlations could be discovered between ***Aluminum*** and ***Calcium*** or ***Sodium***.

Moderate linear correlations either negative or positive are aviable between these pairs: *RI/Si*, *Mg/Al*, *Mg/Ba*, *Ba/Al*, and *RI/Al*.

```
Ion = c("RI", "Na", "Mg", "Al", "Si", "K", "Ca", "Ba", "Fe")    # Ion vector
par(mfrow=c(3,3))
for (i in Ion){
  adjbox(data=Glass, RI ~ Type, xlab="Type", ylab=i, main="Adjusted")}
```



Adjusted box plots shows there is some outliers in data set.Type 2 has a lot of high outliers values. Type 5 has a wide range of values while type 6 and 7 have a very narrow.

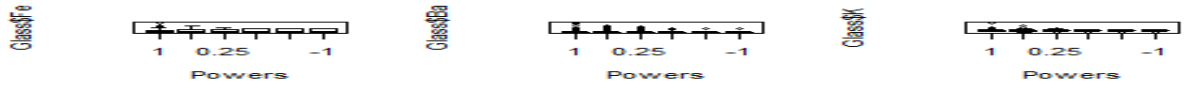
```
apply(Glass[,1:9], 2, skewness)

##          RI          Na          Mg          Al          Si           K
##   1.6254305   0.4541815 -1.1525593   0.9072898 -0.7304472   6.5516483
##          Ca          Ba          Fe
##   2.0470539   3.4164246   1.7543275
```

Skewness for predicors were calculated. It shows that some predictors are highly skewed

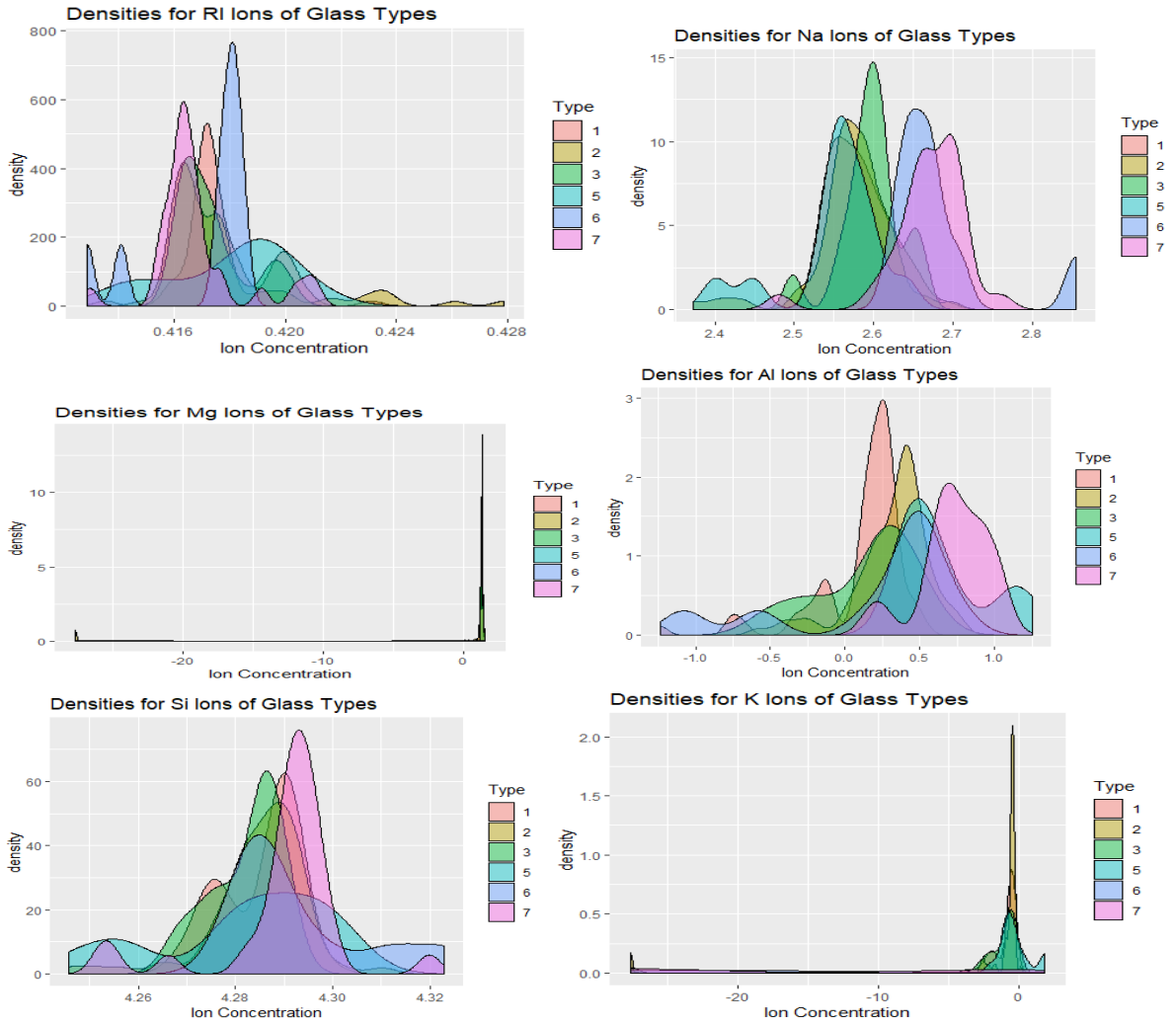
1b)

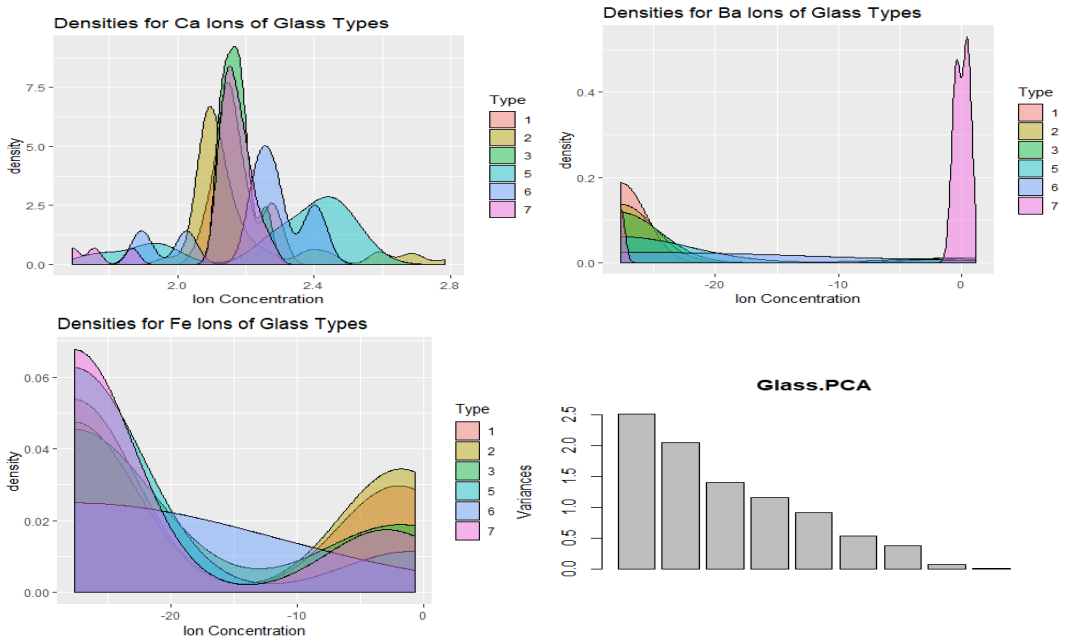
As described above Fe, Ba and K are highly skewed and will benefit from transformation. Because there are zero data there we can produce a very small shift in our data

```
par(mfrow=c(3,3))
symbox(Glass$Fe, start=1e-10, data=Glass, powers=c(1,0.5,0.25,0,-0.5,-1))
symbox(Glass$Ba, start=1e-10, data=Glass, powers=c(1,0.5,0.25,0,-0.5,-1))
symbox(Glass$K, start=1e-10, data=Glass, powers=c(1,0.5,0.25,0,-0.5,-1))
# Reset display

logGlass = log(Glass[,1:9]+1e-12)
logGlass$Type = Glass$Type
# melt the variable
logGlass2 = melt(logGlass)

#Ion2 = c("RI", "Na", "Mg", "Al", "Si", "K", "Ca", "Ba", "Fe")
par(mfrow=c(3,3))
```



```r
for (i in Ion) {
  print(ggplot(logGlass2[logGlass2$variable == i,], aes(x=value, fill=Type))
+
          geom_density(alpha=0.45) +        #set geometry and transparency
          labs(x = "Ion Concentration",          #set x-label and title
               title = paste ("Densities for", i, "Ions of Glass Types")))
}
```






```
lambda.opt <- rep(1, 9)
names(lambda.opt) <- names(Glass[,1:9])
for (i in 2:9) {
  a <- EnvStats::boxcox(Glass[,i]+1e-27, optimize = TRUE, lambda=c(-10,10))
# if optimize = TRUE, then you must tell the function the
  lambda.opt[i] <- a$lambda
}
lambda.opt
```

```
##          RI          Na          Mg          Al          Si           K
##  1.00000000  0.05045353  5.06060897  0.48445306  9.66790955  0.40864153
##          Ca          Ba          Fe
## -0.85935912  0.24494315  0.60028054
```

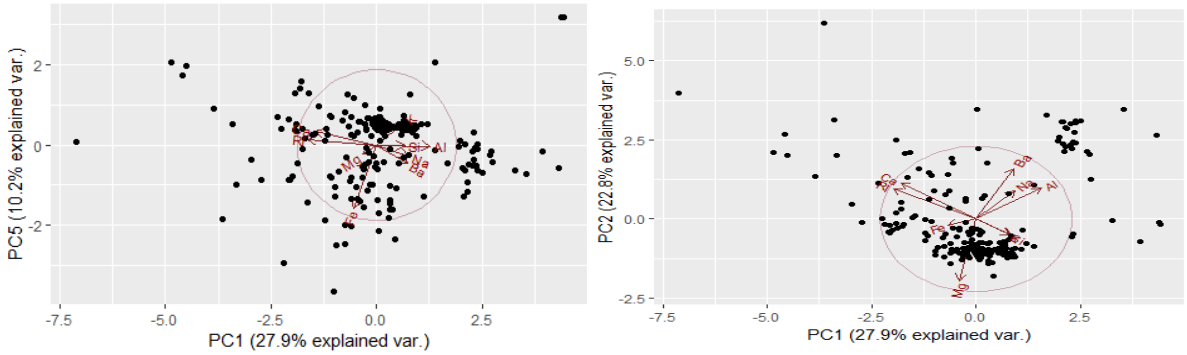
- Optimum values of lambda are numbers with decimal points. It is very important to select a rounded number which could describe the model physically. Noises and erorrs involved in data collection and registration may suggest this values mathematically but I think thatit is better to select more logical and simple model.

(c)

```
Glass.PCA<-prcomp(Glass[,1:9],scale=TRUE)

plot(Glass.PCA)

summary(Glass.PCA)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6    PC7
## Standard deviation     1.585 1.4318 1.1853 1.0760 0.9560 0.72639 0.6074
## Proportion of Variance 0.279 0.2278 0.1561 0.1286 0.1016 0.05863 0.0410
## Cumulative Proportion  0.279 0.5068 0.6629 0.7915 0.8931 0.95173 0.9927
```

```
##                             PC8     PC9
## Standard deviation     0.25269 0.04011
## Proportion of Variance 0.00709 0.00018
## Cumulative Proportion  0.99982 1.00000

par(mfrow=c(1,2))               # Divides the screen into three sections
ggbiplot(Glass.PCA,obs.scale = 1, var.scale = 1, circle = TRUE, choices =
c(1,2))
```




```
ggbiplot(Glass.PCA,obs.scale = 1, var.scale = 1, circle = TRUE, choices =
c(1,5))
```

Pricipal component analysis shows that, we can reduce the dimension from 9 variables into 5 variables keeping about 90% of the variance of data. More investigation on the data and PCs by *ggbiplot* indicates that ***Silisium***, ***Calcium***, and ***Refractive Index*** narates smae story and could be collaped on a single component. They may have different direction but are in same line.

(d)

PCA is an unsupervised method which reduces the problem dimensions keeping data variance as much as possible. On the other hand, LDA is a supervised method which uses the given data to produce a simpler and more accurate model. Here, in our data, table shows that LDA did a very good job in classification of group 7, but its performance was tribble in group 3. For other gruops, the result could acceptale depending on our desire.

Problem 2 :Missing Data

```
# Load data from the package "Amelia"
data(freetrade)
```

(a)

The code for regression using listwise deletetion

(b)

Following code was provided to perform regression after using mean imputation:

```
freetrade.mimp <- freetrade
freetrade.mimp[is.na(freetrade.mimp$tariff), "tariff"] <-
mean(freetrade.mimp$tariff,na.rm=T)
freetrade.mimp[is.na(freetrade.mimp$polity), "polity"] <-
mean(freetrade.mimp$polity,na.rm=T)
freetrade.mimp[is.na(freetrade.mimp$intresmi), "intresmi"] <-
mean(freetrade.mimp$intresmi,na.rm=T)
freetrade.mimp[is.na(freetrade.mimp$signed), "signed"] <-
mean(freetrade.mimp$signed,na.rm=T)
freetrade.mimp[is.na(freetrade.mimp$fiveop), "fiveop"] <-
mean(freetrade.mimp$fiveop,na.rm=T)

freetrade.mimp.fit <- lm(data=freetrade.mimp,
                         tariff ~ year + country + polity + pop + gdp.pc +
intresmi + signed + fiveop + usheg)
```

(C)

Following code was provided to perform regression after using multiple imputation. Different methods such as *mean*, *rf*, *sample*, and *cart*. The lastest one has been used to perform regression.

(d)

The “listwise deletion”" produces best fit for the data though with very low degrees of freedom.For pooled regression using multiple imputation, the countries Korea and Pakistan, Polity and gdp.pc have the most significant coefficients.For pooled regression using single imputation, the countries Pakistan and Thailand, Polity and fiveop have the most significant coefficients. Also,As can be seen by using mean imputation the coeficient would not be changed. However, the coeficient of mice is highly related to the method we are choosing. Imputations based on the modeling use the model as a prediction. They may add some noises to the predicted value, but the reality is that those values are not the missing valuse.I would say **listwise** deletion showed the best results amoung the other methods. Therefore, I would suggest to use this method instead of imputation.

Problem 3: House Price Data :

```
housing <- read.csv("housingData.csv")
```

(a) Transform or construct

find the variables that have missing inputs.

```
na_cols <- which(colSums(is.na(housing)) > 0)
sort(colSums(sapply(housing[na_cols], is.na)), decreasing = TRUE)
```

```
##       PoolQC  MiscFeature        Alley        Fence  FireplaceQu 
##          998          966          938          805          466 
##  LotFrontage   GarageType  GarageYrBlt GarageFinish   GarageQual 
##          207           53           53           53           53 
##   GarageCond BsmtExposure BsmtFinType2     BsmtQual     BsmtCond 
##           53           32           32           31           31 
## BsmtFinType1   MasVnrType   MasVnrArea   Electrical              
##           31            4            4            1
```

Te result shows that the PoolQC feature has a whopping 998 NAs, signifying that these 998 houses do not have a pool. For other features such as Electrical and MasVnrtype or MasVnArea, they only have a few missing inputs with no explanation as to why they are missing, so it's likely that its just an input error. For those features, the most common input is filled in.

Check the nearzero variance

Transformation some variable by using collapse function and getting log

```
housing$LotShape<-fct_collapse(housing$LotShape,
Reg=c("Reg"),IrReg=c("IR1","IR2","IR3"))
housing$Functional<-
fct_collapse(housing$Functional,Type=c("Typ"),deduct=c("Min1","Min2","Mod","M
aj1","Maj2"))
housing$Saleprice<-log(housing$SalePrice)
housing$YearBuilt <- log(housing$YearBuilt)
housing$LotFrontage <- log(housing$LotFrontage)
```

Deleting variable which has so many missing value or redundant variable

```
housing$Alley <-NULL
housing$PoolQC <- NULL
housing$MiscFeature <- NULL
housing$Fence <- NULL
housing$BsmtQual <- NULL
housing$Exterior1st <- NULL
housing$ExterQual <- NULL
```

Converiting some categorical to numerical for feature engeering

```
housing$OverallQual <- as.numeric(housing$OverallQual)
housing$OverallCond <- as.numeric(housing$OverallCond)
housing$GarageCond <- as.numeric(housing$GarageCond)
housing$GarageQual <- as.numeric(housing$GarageQual)
housing$BsmtFinType1 <- as.numeric(housing$BsmtFinType1)
housing$BsmtFinType2 <- as.numeric(housing$BsmtFinType2)
```

Combine some variables that have the same concept with each other through multypling or summing them

```
housing$OverallGrade <- (housing$OverallQual) * (housing$OverallCond)
housing$GarageGrade <- (housing$GarageQual) * (housing$GarageCond)
housing$BasementRating <- (housing$BsmtFinType1) * (housing$BsmtFinType2)
```

```
housing$TotalBath <- (housing$BsmtFullBath) + (0.5 * housing$BsmtHalfBath) +
(housing$FullBath) + (0.5 * housing$HalfBath)
housing$TotalSF <- (housing$TotalBsmtSF) + (housing$BsmtFinSF1) +
(housing$BsmtFinSF2)+ (housing$X1stFlrSF) + (housing$X2ndFlrSF)
housing$TotalPorchSF <- (housing$OpenPorchSF) + (housing$EncPorchSF)

housing <- housing[,-c(14,15,26,27,28,29,30,35,36,39,40,41,42,59,60)]
```

(b) Jutify the choices made in feature engineering

For feature engineering first I tryied to find missing Value and check the nearzero variannce then decide to delete some variable like“Fence”,“PoolQc”,“Alley” and “Miscfeature” beacuse of the majorty of their data was not avalilabe. Then convert some categorical variable to numeric for better result of feature enginerring and construct new variable. After that I tried to multyple or sume some varibale of the data set which has they repeated alot like variables taht have both related to each other and have same condtion and quality. Or sum variable to gain the totall. Also I got log from some variable to make them like normal ditribution and collapse some variable beacuse they have many uninformative Level. All I did help to better undrestand the data by deleting,reduceing or combining unnececery vaiable. below you can seen that how some variable by getting log their distribution changed

```
par(mfrow=c(1,3))
hist(log(housing$Saleprice))
hist(log(housing$LotFrontage))
hist(log(housing$SalePrice))
```