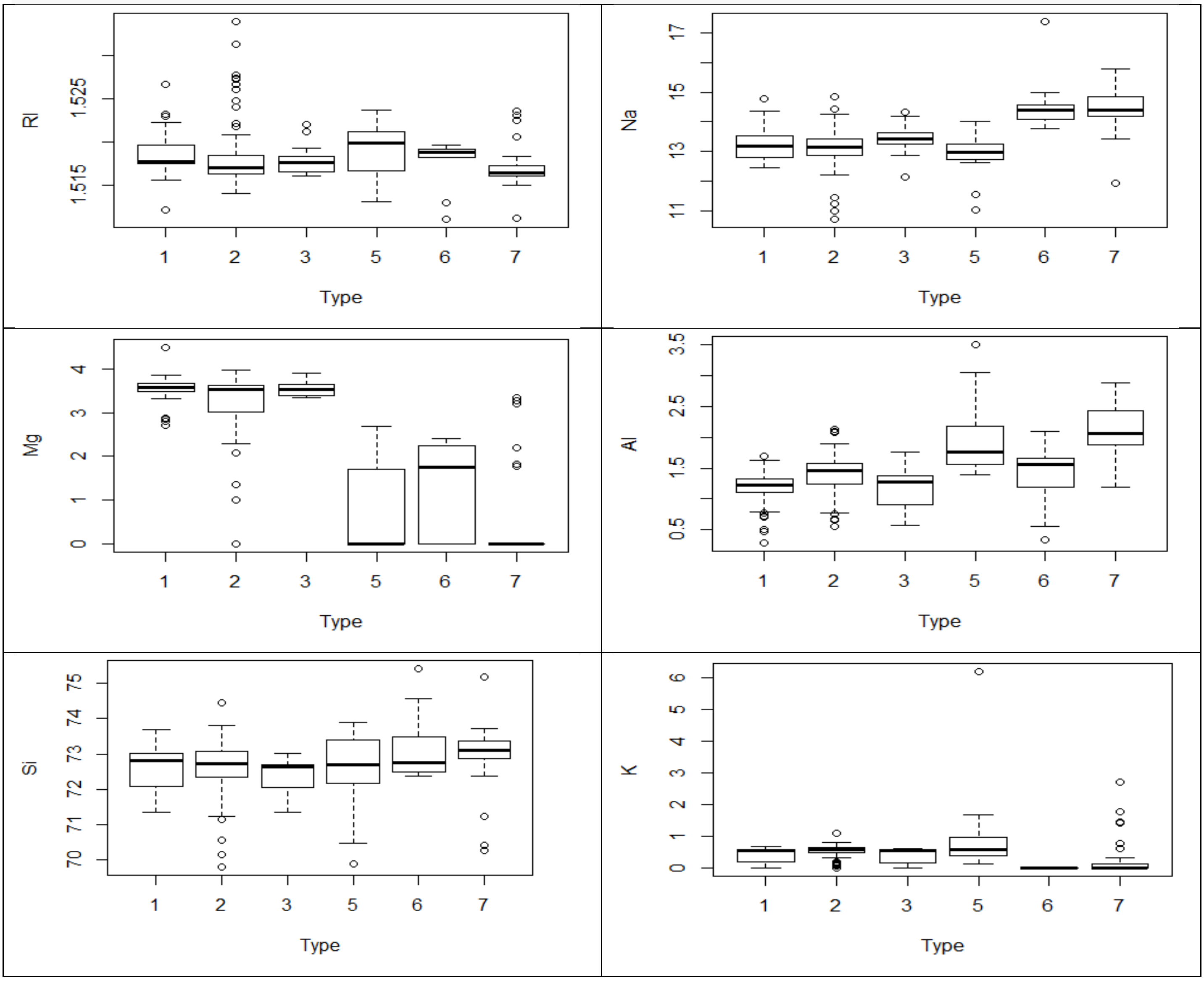
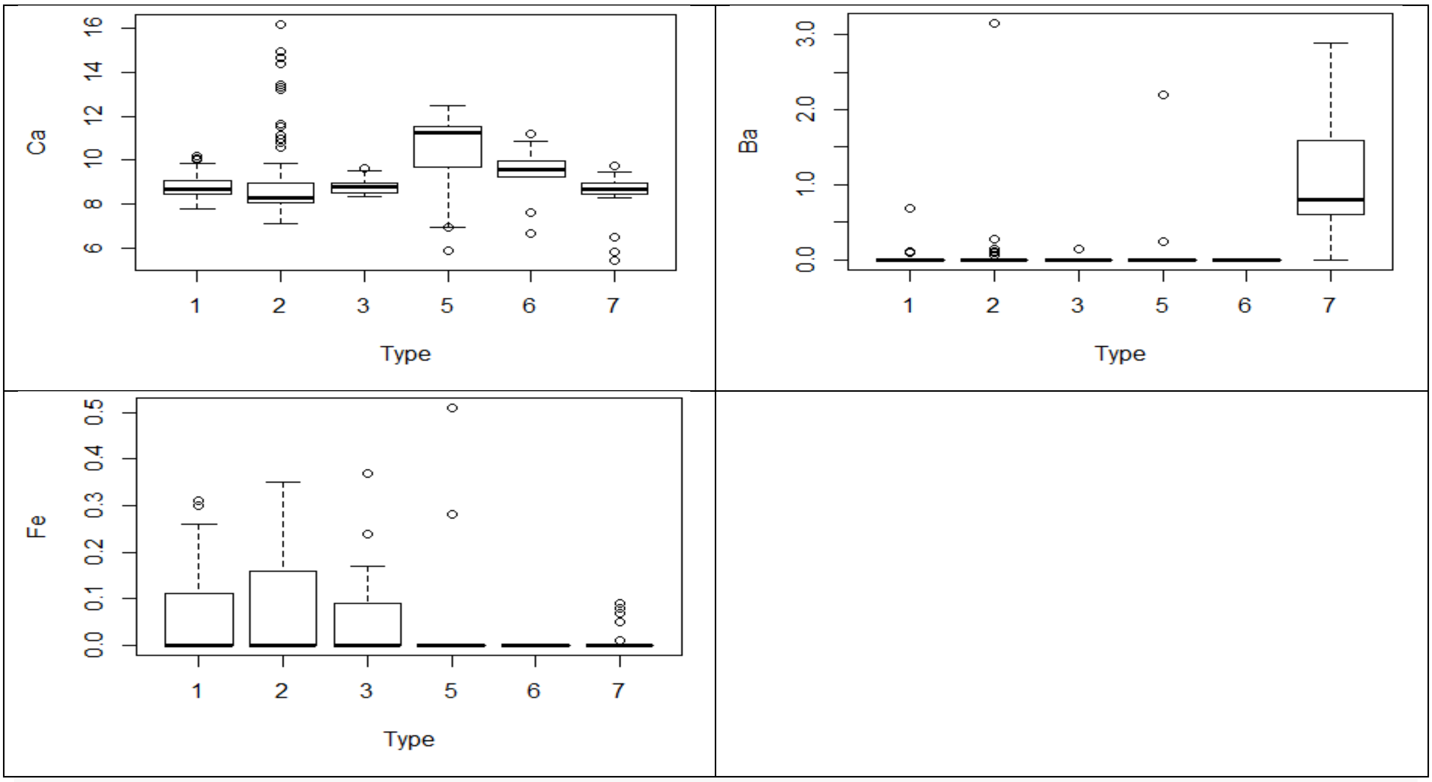
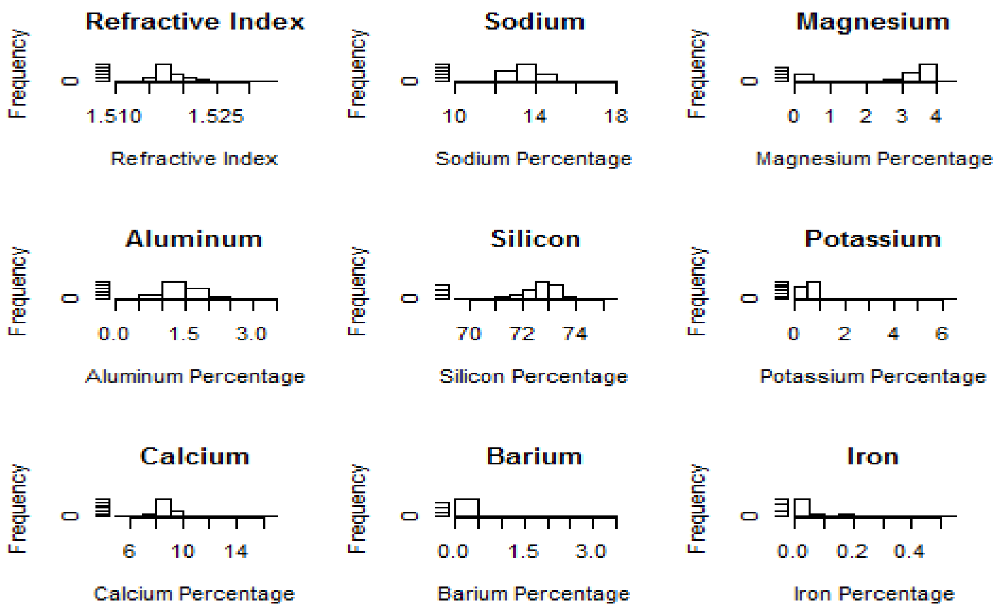
```r
#==============================================================================
#                            Using R: Glass Identification
#==============================================================================
#The study of classification of types of glass is motivated by criminological ...
library(mlbench)
data(Glass)

#------------------------------------------------------------------------------
#(a) Using visualizations, explore the predictor variables to understand their ...
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")
```



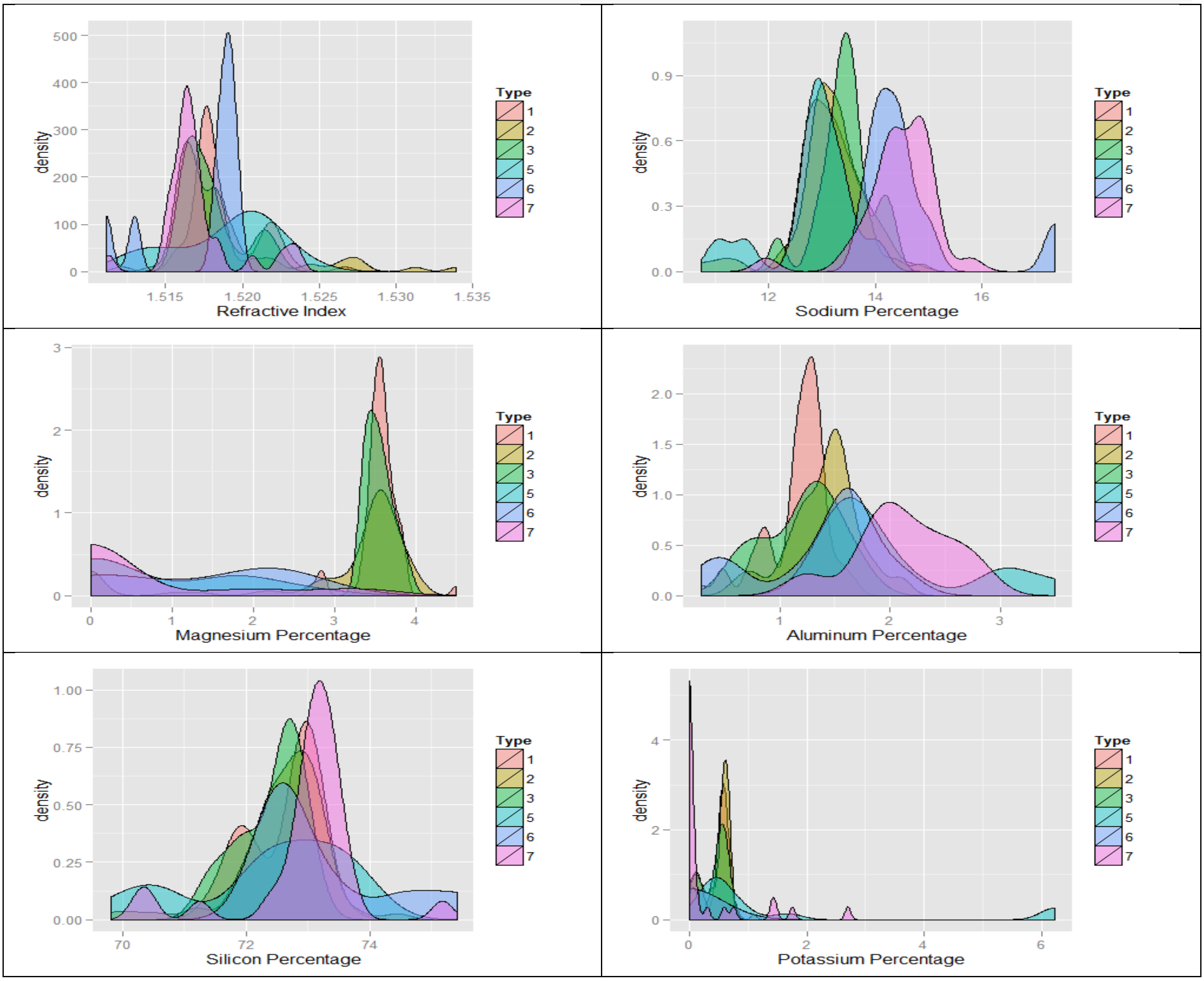


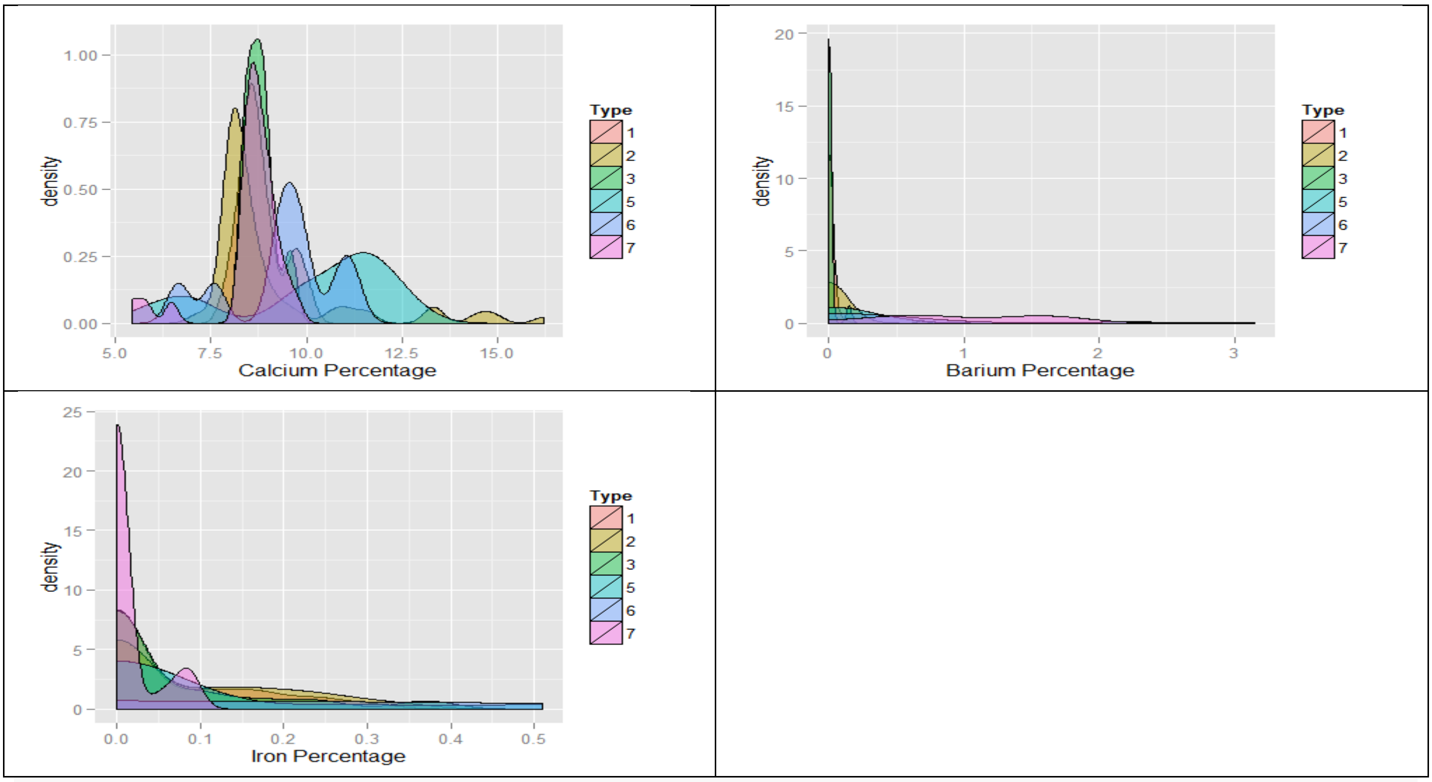

```
ar(mfrow=c(3,3))
hist(Glass$RI, main="Refractive Index", xlab="Refractive Index")
hist(Glass$Na, main="Sodium", xlab="Sodium Percentage")
hist(Glass$Mg, main="Magnesium", xlab="Magnesium Percentage")
hist(Glass$Al, main="Aluminum", xlab="Aluminum Percentage")
hist(Glass$Si, main="Silicon", xlab="Silicon Percentage")
hist(Glass$K,  main="Potassium", xlab="Potassium Percentage")
hist(Glass$Ca, main="Calcium", xlab="Calcium Percentage")
hist(Glass$Ba, main="Barium", xlab="Barium Percentage")
hist(Glass$Fe, main="Iron", xlab="Iron Percentage")
```



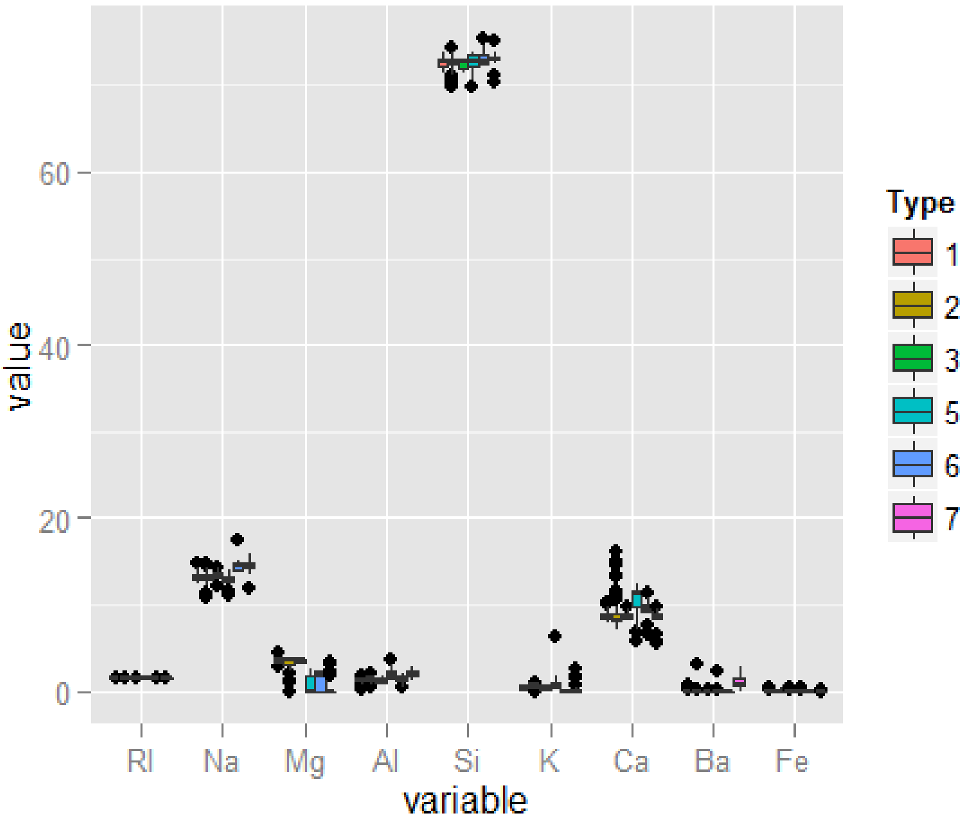

```
par(mfrow=c(1,1))

library(reshape2)
Glass.melt<- melt(Glass)

library(ggplot2)
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")
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")
```

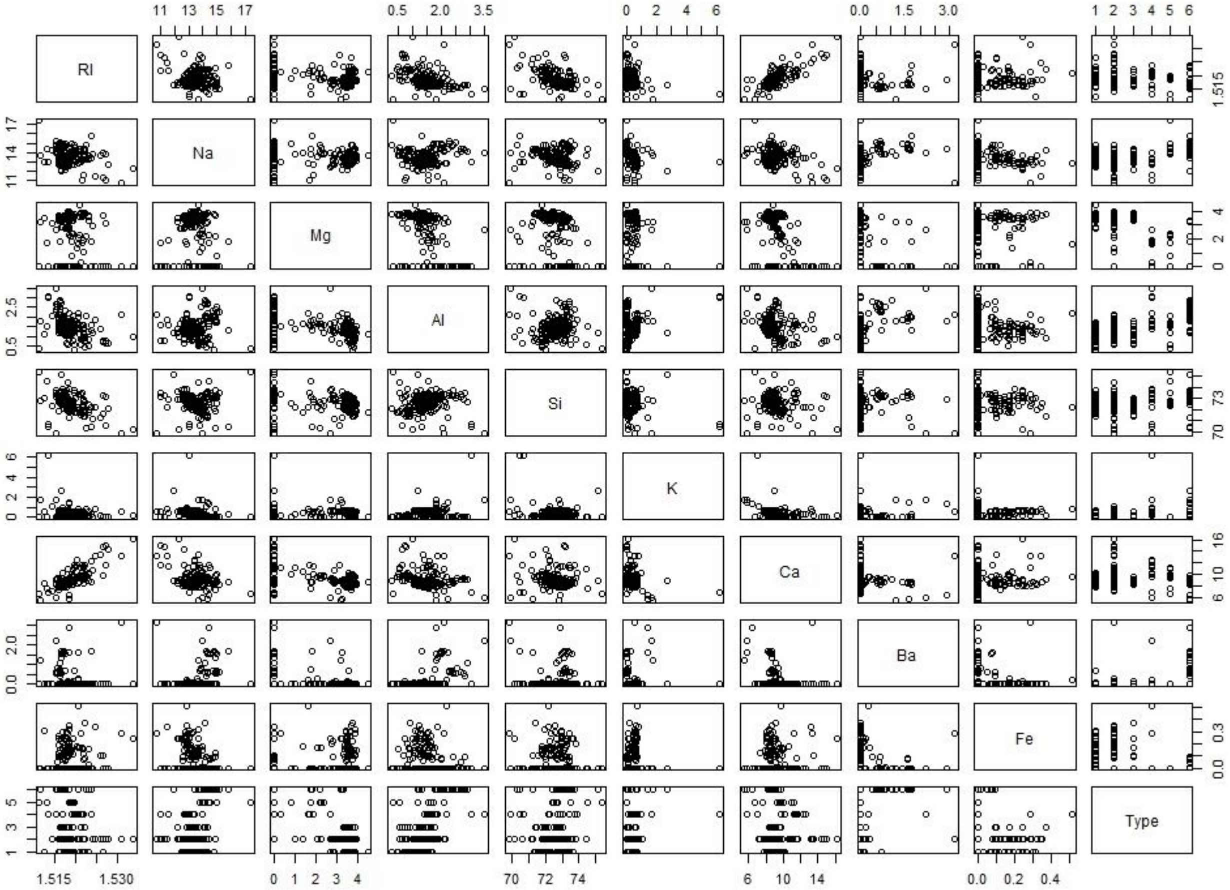




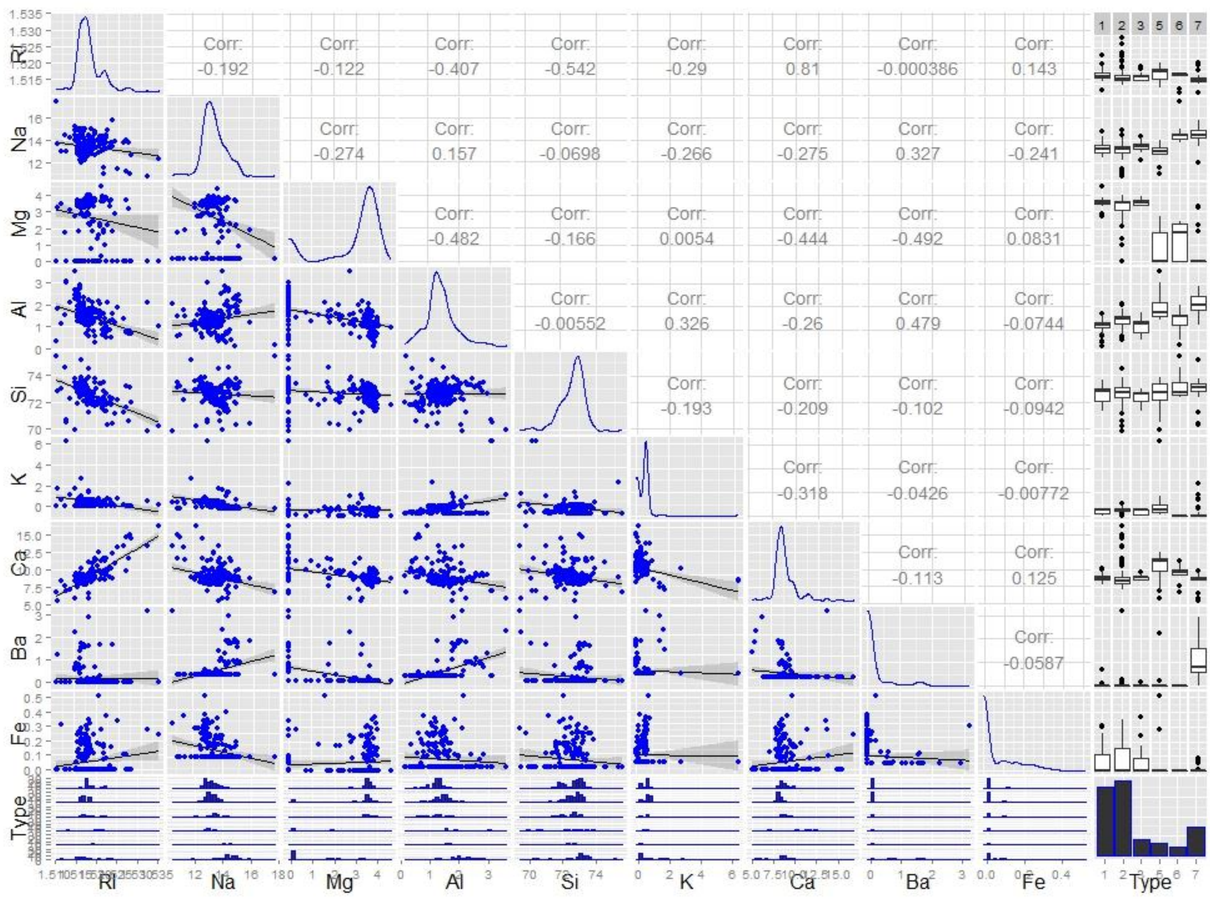

```
ggplot(Glass.melt,aes(x=variable, y=value, fill=Type)) + geom_boxplot()
```



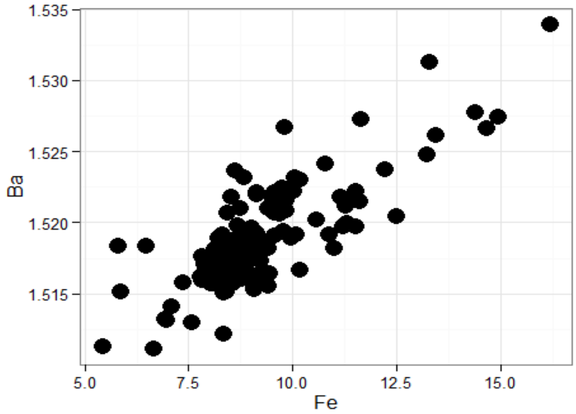

```
plot(Glass)
```



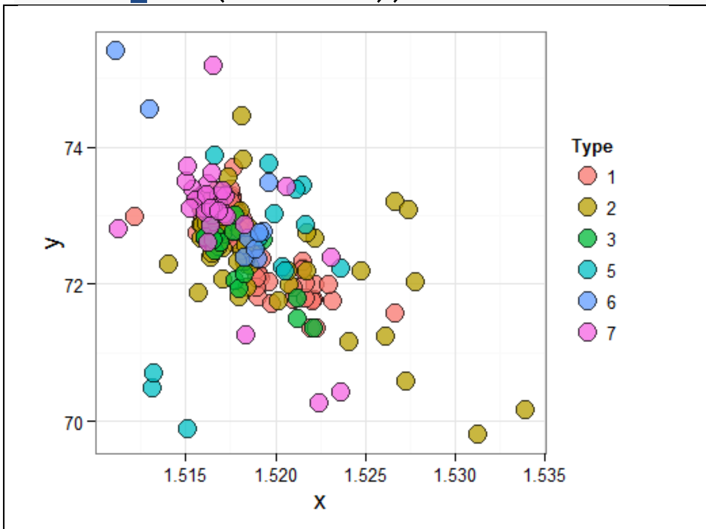

```r
library(GGally)
ggpairs(Glass, lower=list(continuous="smooth", params=c(colour="blue")),
        diag=list(params=c(colour="blue")), upper=list(params=list(corSize=6)),
axisLabels='show')
```

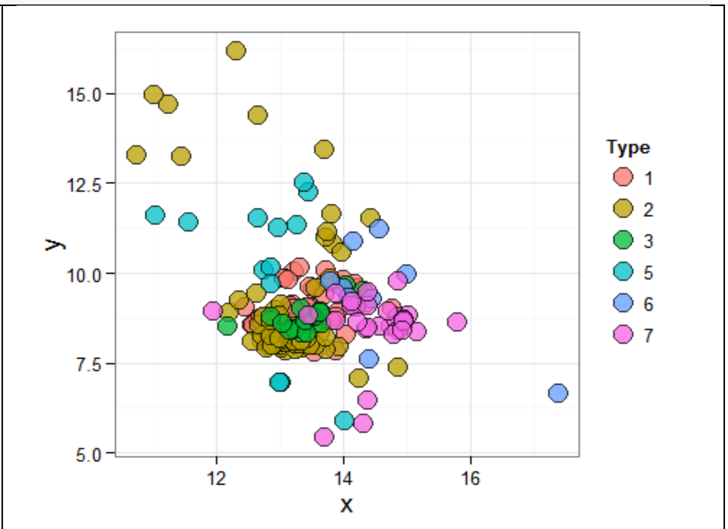


```r
#Considering the general trend, more detailed graph of scatter plot can be plotted as follow:
qplot(data=Glass, x=Ca, y=RI, size=I(5)) + theme_bw() + labs(y = "Ba", x = "Fe")
```

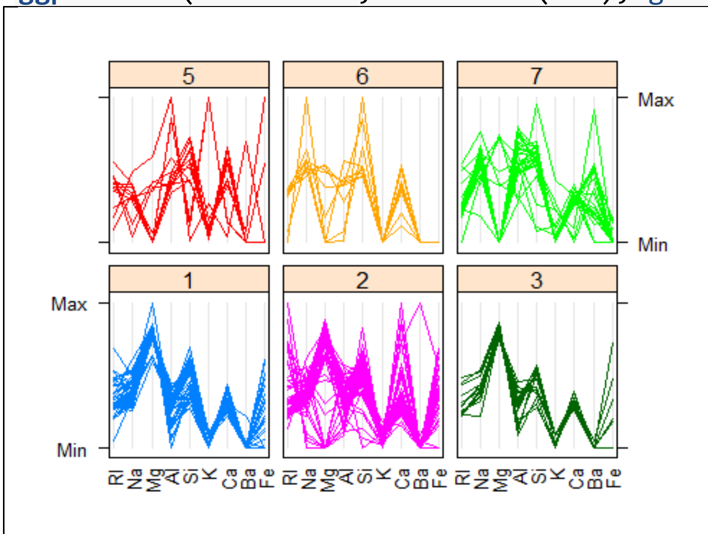


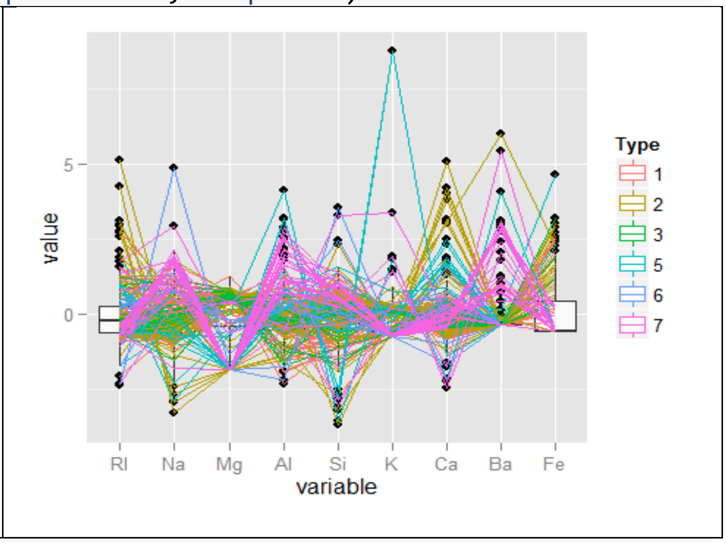

```r
ggplot(data=Glass, aes(x=RI,y=Si))+ geom_point(aes(fill=Type), alpha=I(.75), colour="black", pch=21, size=5)+
  theme_bw()+ labs(y="y", x="x") + theme(legend.key=element_blank(), axis.title = element_text(size = 14))
ggplot(data=Glass, aes(x=Na,y=Ca))+ geom_point(aes(fill=Type), alpha=I(.75), colour="black", pch=21, size=5)+
  theme_bw()+ labs(y="y", x="x") + theme(legend.key=element_blank(), axis.title = element_text(size = 14))
```



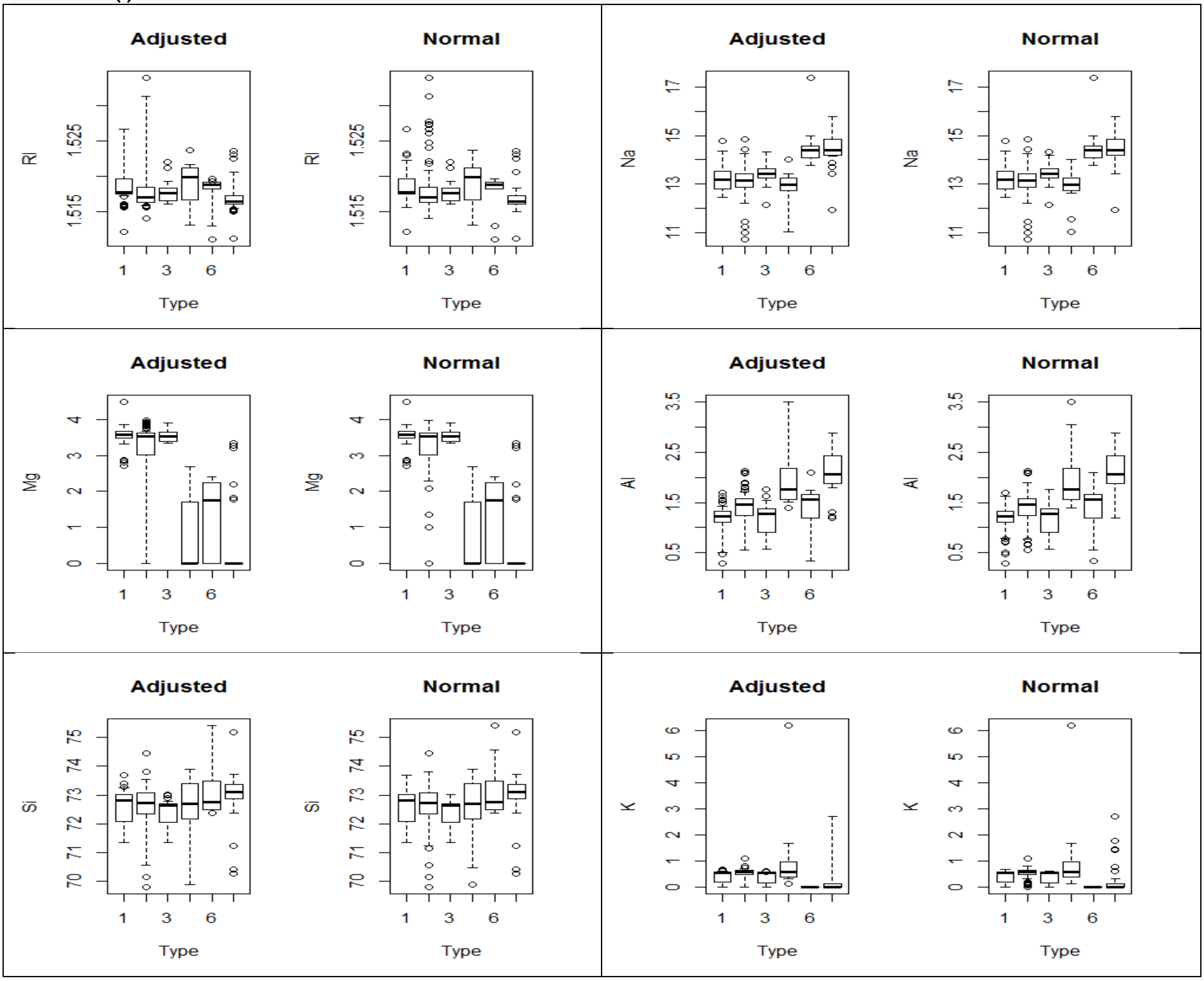


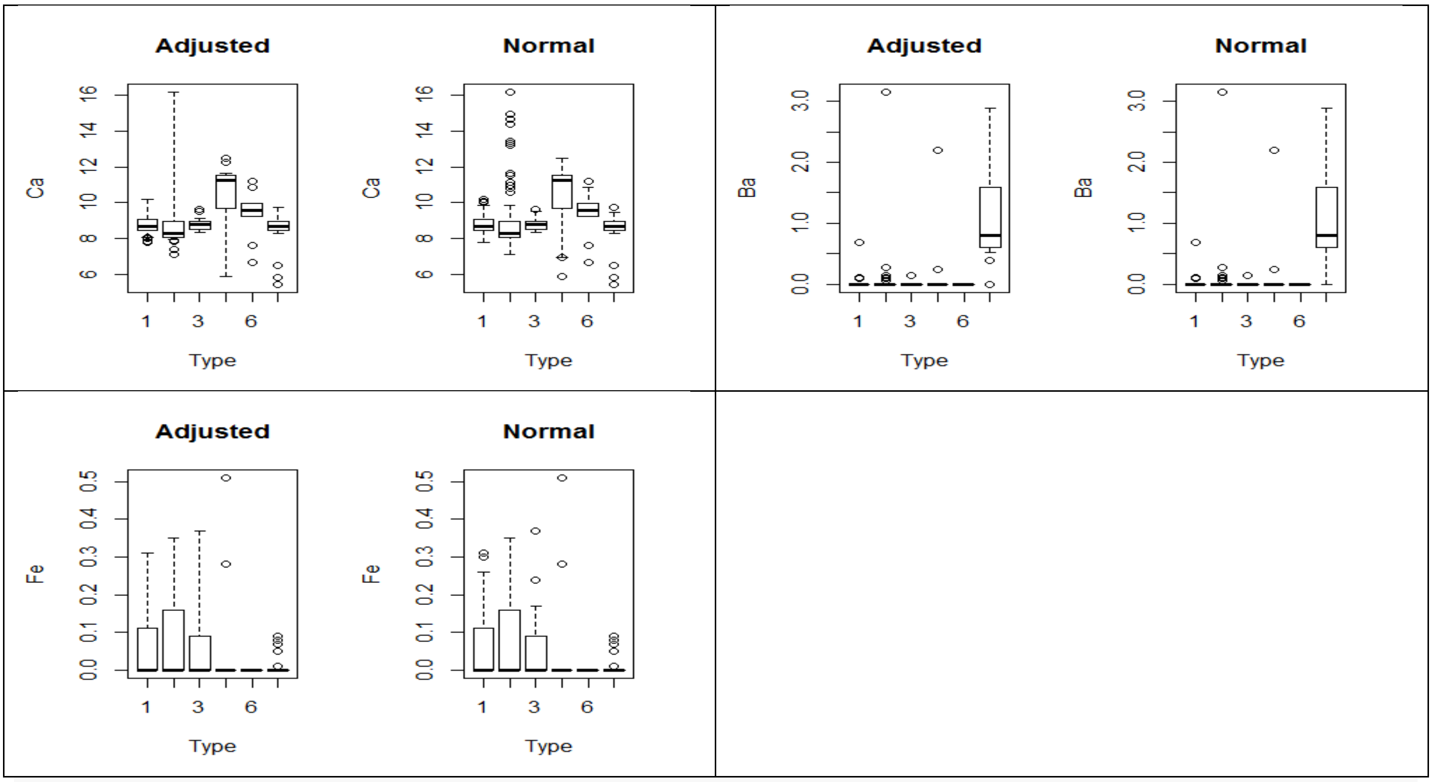

```r
library(lattice)
parallelplot(~Glass[1:9] | Type, data=Glass, groups = Type, horizontal.axis = FALSE, scales = list(x = list(rot = 90)))
ggparcoord(data=Glass, columns=c(1:9), groupColumn=10, boxplot=T)
```





```
library(robustbase)
par(mfrow=c(1,2))
adjbox(data=Glass, RI ~ Type, xlab="Type", ylab="RI", main="Adjusted")
boxplot(data=Glass, RI ~ Type, xlab="Type", ylab="RI", main="Normal")
adjbox(data=Glass, Na ~ Type, xlab="Type", ylab="Na", main="Adjusted")
boxplot(data=Glass, Na ~ Type, xlab="Type", ylab="Na", main="Normal")
adjbox(data=Glass, Mg ~ Type, xlab="Type", ylab="Mg", main="Adjusted")
boxplot(data=Glass, Mg ~ Type, xlab="Type", ylab="Mg", main="Normal")
adjbox(data=Glass, Al ~ Type, xlab="Type", ylab="Al", main="Adjusted")
boxplot(data=Glass, Al ~ Type, xlab="Type", ylab="Al", main="Normal")
adjbox(data=Glass, Si ~ Type, xlab="Type", ylab="Si", main="Adjusted")
boxplot(data=Glass, Si ~ Type, xlab="Type", ylab="Si", main="Normal")
adjbox(data=Glass, K ~ Type, xlab="Type", ylab="K", main="Adjusted")
boxplot(data=Glass, K ~ Type, xlab="Type", ylab="K", main="Normal")
adjbox(data=Glass, Ca ~ Type, xlab="Type", ylab="Ca", main="Adjusted")
boxplot(data=Glass, Ca ~ Type, xlab="Type", ylab="Ca", main="Normal")
adjbox(data=Glass, Ba ~ Type, xlab="Type", ylab="Ba", main="Adjusted")
boxplot(data=Glass, Ba ~ Type, xlab="Type", ylab="Ba", main="Normal")
adjbox(data=Glass, Fe ~ Type, xlab="Type", ylab="Fe", main="Adjusted")
boxplot(data=Glass, Fe ~ Type, xlab="Type", ylab="Fe", main="Normal")
par(mfrow=c(1,1))
dev.off()
```






```
cor(Glass[,1:9],method = "kendall")
##              RI           Na          Mg          Al          Si
## RI  1.00000000  0.032005325  0.10409058 -0.35566293 -0.39475299
## Na  0.03200533  1.000000000 -0.07048265  0.07718298 -0.22554402
## Mg  0.10409058 -0.070482645  1.00000000 -0.38229174 -0.24913827
## Al -0.35566293  0.077182982 -0.38229174  1.00000000  0.14151373
## Si -0.39475299 -0.225544018 -0.24913827  0.14151373  1.00000000
## K  -0.23081766 -0.445621050  0.12422291  0.13988194  0.01748374
## Ca  0.52821355  0.003879557 -0.21356262 -0.22112377 -0.15271012
## Ba -0.14001101  0.322975259 -0.36703875  0.36720246  0.13583205
## Fe  0.07179747 -0.172392101  0.07095252 -0.05774051 -0.05497821
cor(Glass[,1:9],method = "pearson")
cor(Glass[,1:9],method = "spearman")
#As can be seen from the exploration, especially from the ggpairs (correlation) it seems there are some relationship between some of the predictors such as positive relationship between RI and Ca, negative relationship between RI and Si and etc.
#As can be seen different method of calculating correlation have different results. However, they all agree on some too. All show the positive correlation between RI and Ca, Na and Ba or Negative correlation between Mg and Al. It seems there are some outliers especially when they are categorized based on the glass type.
#One can use boxplot and adjusted boxplot to recognize potential outliers. Using some test can help to recognize outliers but the distribution of some of the predictors are not normal and need to be transformed.
#There are some highly skewed distribution on Fe, Ba and K.
#------------------------------------------------------------------------------------
#(b) Identify three attributes that you think could benefit from a skew ...
#i. Use the symbox function from package car to consider possible power transformations
library(car)

#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
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))
```

```
#ii. Use the boxcox method from the EnvStats package to determine an optimal ...
library(EnvStats)
c <- boxcox(Glass$Fe+1e-10, optimize=TRUE, lambda=c(-10,10))
c[1]
lambda=0.6002792
c <- boxcox(Glass$Ba+1e-10, optimize=TRUE, lambda=c(-10,10))
c[1]
lambda=0.2435362
c <- boxcox(Glass$K+1e-10, optimize=TRUE, lambda=c(-10,10))
c[1]
lambda=0.4084975
#-------------------------------------------------------------------------------------
#(c) Use PCA to help evaluate the data. Does this provide any insight? If so, what?
PCA.Glass <- prcomp(Glass[,1:9], scale=T)
library(ggbiplot)
ggbiplot(PCA.Glass, circle=T, choices=c(1,2), obs.scale=1, varname.size=7)
plot(PCA.Glass)
summary(PCA.Glass)
## Importance of components:
##                          PC1    PC2    PC3    PC4    PC5     PC6    PC7
## Standard deviation     1.585 1.4318 1.1853 1.0760 0.9560 0.72639 0.6074
## Proportion of Variance 0.279 0.2278 0.1561 0.1286 0.1016 0.05863 0.0410
## Cumulative Proportion  0.279 0.5068 0.6629 0.7915 0.8931 0.95173 0.9927
##                            PC8     PC9
## Standard deviation     0.25269 0.04011
## Proportion of Variance 0.00709 0.00018
## Cumulative Proportion  0.99982 1.00000
#Using only 5 PCs we can preserve almost 90% of our data. So it means we can reduce
our Dimentions to 5 with high accuracy.
#The cumulative proportion of the PCA for the first two PCs is about 50%. Here the Mg,
RI, Ca, Ba, Al have the
#most influence and Na is medium and Fe, K and Si have very small contribution.
#-------------------------------------------------------------------------------------
#(d) Perform a linear discriminant analysis (LDA) ...
library(MASS)
fit <- lda(Glass$Type ~ Glass$RI + Glass$Na + Glass$Mg + Glass$Al + Glass$Si + Glass$K
+ Glass$Ca + Glass$Ba + Glass$Fe, data=Glass, CV=F)
fit
fit.predict <- predict(fit, newdata=Glass[,1:9])$class
table(fit.predict, Glass[,10])
##
## fit.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
#Using Linear Discriminate Analysis, it can be seen that it has some error predicting
the type for example it can only say type 1 correctly in 52 cases out of 70 cases. or
it say in 17 cases that the type is 1 considering that it is actually type 2. Or it
cannot predict type 3 in any observation. But it can predict glass type 7 very good.
#So it depends on the accuracy we are seeking to help us accept or reject LDA for this
data. PCA tries to transform data and can be used to reduce the dimension while LDA
tries to discriminate the data. In my opinion both of them should be used together
since they are following different scenarios.
```