KJ_Chapter3_Problem1

Jonathan Hernandez

3.1. The UC Irvine Machine Learning Repository 6 contains a data set related to glass identification. The data consist of 214 glass samples labeled as one of seven class categories. There are nine predictors, including the refractive index and percentages of eight elements: Na, Mg, Al, Si, K, Ca, Ba, and Fe. The data can be accessed via:
'library(mlbench)'
'data(Glass)'
'str(Glass)'

(a) Using visualizations, explore the predictor variables to understand their distributions as well as the relationships between predictors.

(b) Do there appear to be any outliers in the data? Are any predictors skewed?

(c) Are there any relevant transformations of one or more predictors that might improve the classification model?

- Let's load the dataset as requested and examine some plots.

```
library(mlbench)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(grid)
library(gridExtra) # grid.arrange for arranging plots
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
data("Glass")
str(Glass)
```

```
## 'data.frame':    214 obs. of  10 variables:
##  $ RI  : num  1.52 1.52 1.52 1.52 1.52 ...
##  $ Na  : num  13.6 13.9 13.5 13.2 13.3 ...
##  $ Mg  : num  4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...
##  $ Al  : num  1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...
##  $ Si  : num  71.8 72.7 73 72.6 73.1 ...
##  $ K   : num  0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...
##  $ Ca  : num  8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...
##  $ Ba  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Fe  : num  0 0 0 0 0 0.26 0 0 0 0.11 ...
##  $ Type: Factor w/ 6 levels "1","2","3","5",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(Glass)
```

```
##        RI              Na              Mg              Al        
##  Min.   :1.511   Min.   :10.73   Min.   :0.000   Min.   :0.290  
##  1st Qu.:1.517   1st Qu.:12.91   1st Qu.:2.115   1st Qu.:1.190  
##  Median :1.518   Median :13.30   Median :3.480   Median :1.360  
##  Mean   :1.518   Mean   :13.41   Mean   :2.685   Mean   :1.445  
##  3rd Qu.:1.519   3rd Qu.:13.82   3rd Qu.:3.600   3rd Qu.:1.630  
##  Max.   :1.534   Max.   :17.38   Max.   :4.490   Max.   :3.500  
##        Si              K                Ca              Ba        
##  Min.   :69.81   Min.   :0.0000   Min.   : 5.430   Min.   :0.000  
##  1st Qu.:72.28   1st Qu.:0.1225   1st Qu.: 8.240   1st Qu.:0.000  
##  Median :72.79   Median :0.5550   Median : 8.600   Median :0.000  
##  Mean   :72.65   Mean   :0.4971   Mean   : 8.957   Mean   :0.175  
##  3rd Qu.:73.09   3rd Qu.:0.6100   3rd Qu.: 9.172   3rd Qu.:0.000  
##  Max.   :75.41   Max.   :6.2100   Max.   :16.190   Max.   :3.150  
##        Fe          Type    
##  Min.   :0.00000   1:70  
##  1st Qu.:0.00000   2:76  
##  Median :0.00000   3:17  
##  Mean   :0.05701   5:13  
##  3rd Qu.:0.10000   6: 9  
##  Max.   :0.51000   7:29
```

- Using ggplot2 with geom_histogram to see the distribution and using box-pairs. Make a variable to hold the plot object of each predictor variable and use grid.arrange to show a 3x3 layout

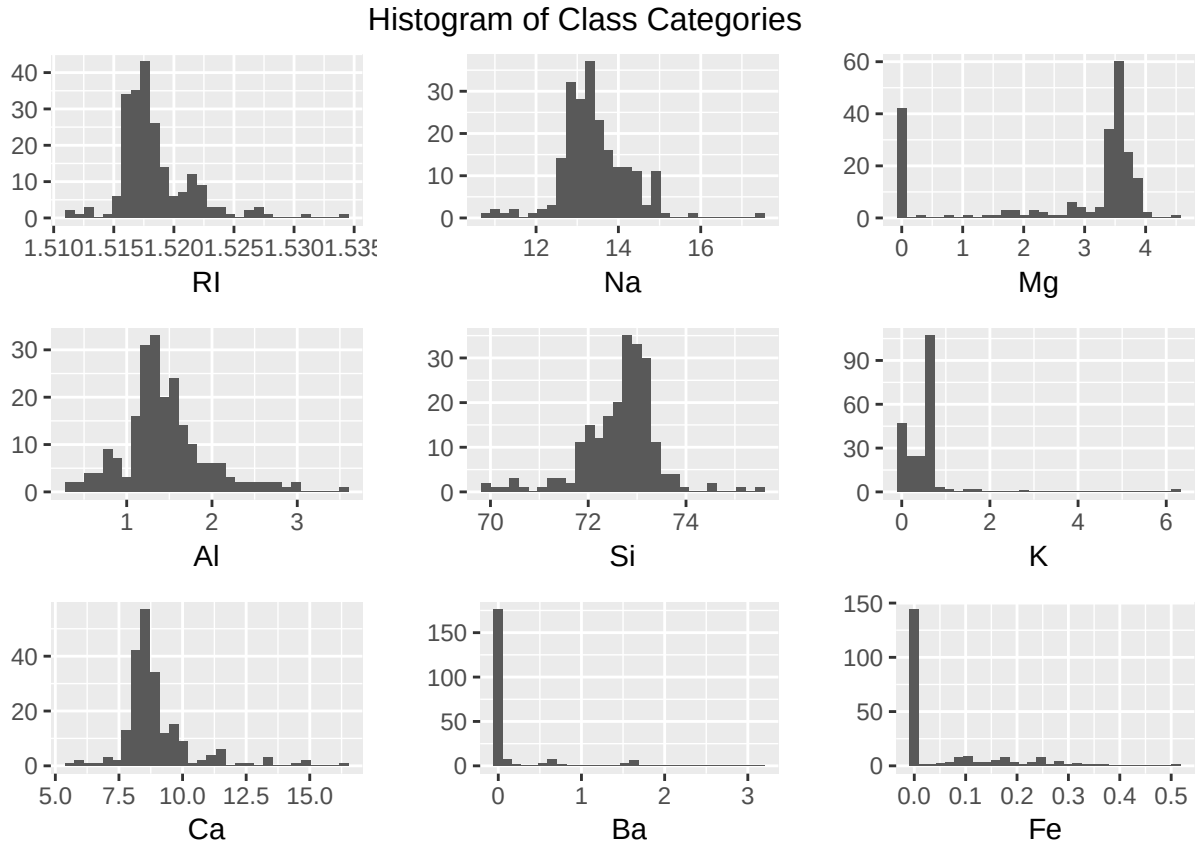
```
# each predictor variable
RI_hist <- ggplot(Glass,aes(RI)) + geom_histogram() + ylab("")
Na_hist <- ggplot(Glass,aes(Na)) + geom_histogram() + ylab("")
Mg_hist <- ggplot(Glass,aes(Mg)) + geom_histogram() + ylab("")
Al_hist <- ggplot(Glass,aes(Al)) + geom_histogram() + ylab("")
```

```
Si_hist <- ggplot(Glass,aes(Si)) + geom_histogram() + ylab("")
K_hist <- ggplot(Glass,aes(K)) + geom_histogram() + ylab("")
Ca_hist <- ggplot(Glass,aes(Ca)) + geom_histogram() + ylab("")
Ba_hist <- ggplot(Glass,aes(Ba)) + geom_histogram() + ylab("")
Fe_hist <- ggplot(Glass,aes(Fe)) + geom_histogram() + ylab("")

# make a 3x3 grid to show each one in a single plot.

grid.arrange(RI_hist, Na_hist, Mg_hist,
             Al_hist, Si_hist, K_hist,
             Ca_hist, Ba_hist, Fe_hist, ncol=3, nrow=3,
             top = textGrob("Histogram of Class Categories"))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

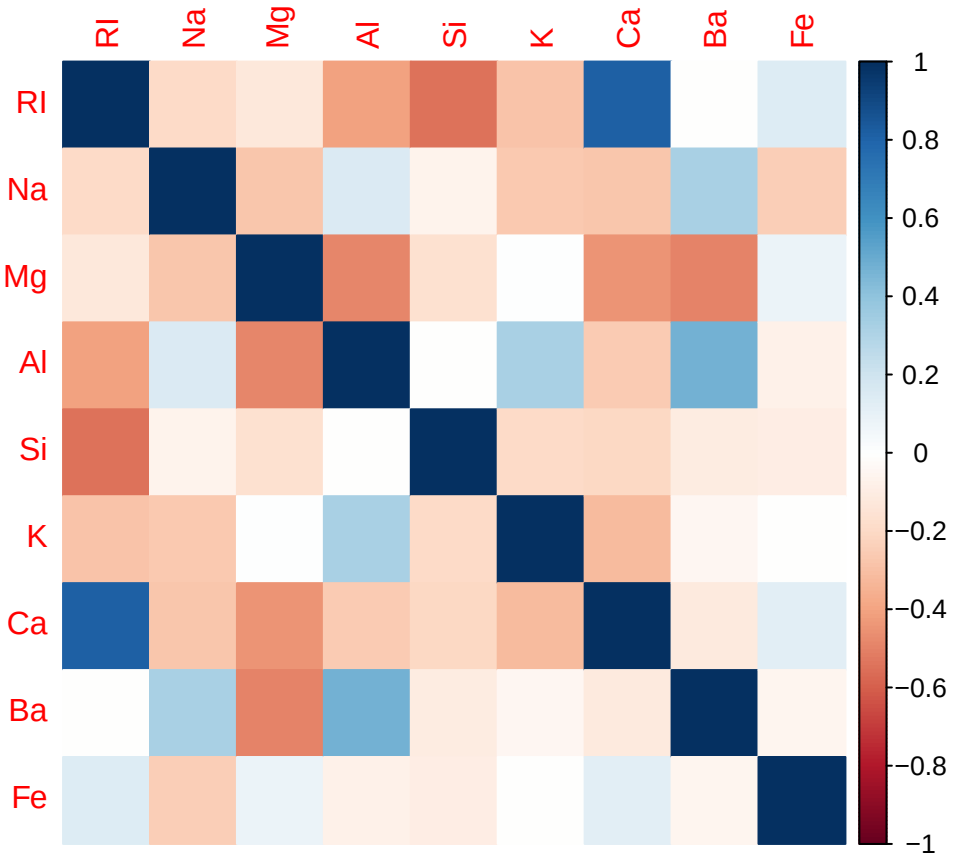


- In these histograms of predictor variables, we see that several variables are skewed. We also see that

K, Ba and Fe follow quite a heavy right skewed distribution. The other variables besides Mg follow similar distributions as well.

- Let's see the relationship of these variables using the corrplot package which will help us visualize the correlation matrix among these variables.

```
corr_matrix <- cor(Glass[,-10]) # matrix to two decimal places
corrplot(corr_matrix, method = "color")
```



- Looking the correlation plot
- Fe, Ba, doesn't have strong correlation amongst the other variables.
- Ca and RI have the strongest correlation (positive)
- There are more negative weak correlations between predictors than any other correlation.
- Adding the correlation with the response variable "Type". The matrix will tell us how well related the predictors are to the response variable.

```
cor(Glass[,-10], as.numeric(Glass$Type))
```

```
##            [,1]
## RI -0.168739357
## Na  0.506424080
## Mg -0.728159518
```

```
## Al  0.591197598
## Si  0.149690687
## K  -0.025834560
## Ca -0.008997841
## Ba  0.577676375
## Fe -0.183206747
```

- From the above column, Mg, Al and Ba have the highest correlation in regards to glass Type. Mg has a strong negative relationship and Al and Ba have pretty strong positive relationships.

b) We see from the histograms above from a) that many of the variables have skewness some more than others. Furthermore, there are several outliers by looking at the histogram. I would expect there to be skewness as each glass sample can have random percentages of the elements.

c) Let's first transform the predictors Mg, K, Ba and, Fe using a box-cox transform as they are heavily skewed with large tails. I will use the BoxCoxTrans() function to achieve this so I can get the optimum lambda value for each predictor. One thing from the summary is that these predcitors I'm choosing have a min value of 0 so it will help to add a small amount say 1e-6 (0.000001) to make the function work.

```
library(caret) # for boxcoxtrans function
```

```
## Loading required package: lattice
```

```
library(e1071) # for skewness function
# Box-cox transforms
Glass$K <- Glass$K + 1e-6
Glass$Ba <- Glass$Ba + 1e-6
Glass$Fe <- Glass$Fe + 1e-6
Glass$Mg <- Glass$Mg + 1e-6
```

- After transforming the variables, we can then apply the Box-Cox transform to all the predictor variables and see the histogram of predictors. We can also see the skewness before and after the transformations.

```
library(e1071)
# function to use the boxcoxtrans function and predict after transforming and
# computing skewness
boxcoxplots <- function(class){
  boxcox <- BoxCoxTrans(class)
  boxcox_pred <- predict(boxcox, class)
  # find the skewness after using the box-cox transform
  skewness(boxcox_pred)
}
```

- Let's do a before and after of skewness of each predictor to see if using the box-cox transform fixed the skewness

```
# before
apply(Glass[,-10], 2, skewness)
```

```
##         RI         Na         Mg         Al         Si          K
##  1.6027151  0.4478343 -1.1364523  0.8946104 -0.7202392  6.4600889
##         Ca         Ba         Fe
##  2.0184463  3.3686800  1.7298107
```

```
# after
apply(Glass[,-10], 2, boxcoxplots)
```

```
##          RI          Na          Mg          Al          Si           K
##  1.56566039  0.03384644 -1.43270870  0.09105899 -0.65090568 -0.78216211
##          Ca          Ba          Fe
## -0.19395573  1.67566612  0.74424403
```

- The skewness was reduced to be slightly skewed instead of before, doing this transformation on variables especially like K, Ba, Fe and Ca and Mg might make predicting glass type better.