DATA 624 - Homework 4

OMER OZEREN

Table of Contents

Exercises 3.1 and 3.2 from the Kuhn and Johnson book “Applied Predictive Modeling”.

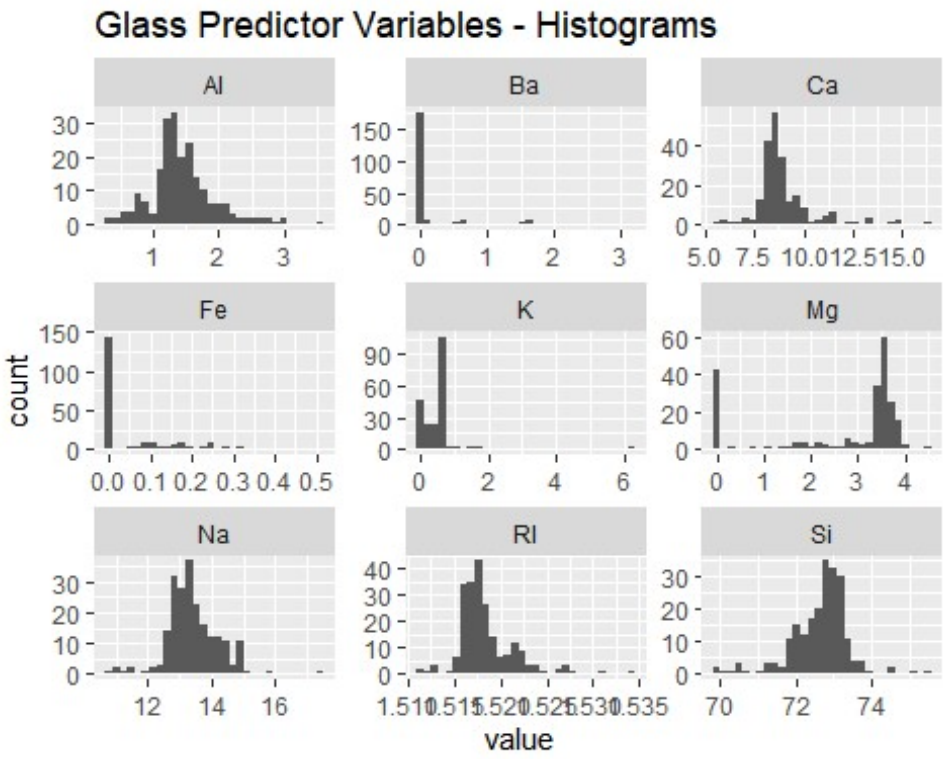
```
#clear the workspace
rm(list = ls())
#load req's packages
library(mlbench)
library(ggplot2)
library(GGally)
library(dplyr)
library(corrplot)
library(tidyr)
library(psych)
library(knitr)
library(DMwR)
```

Question 3.1

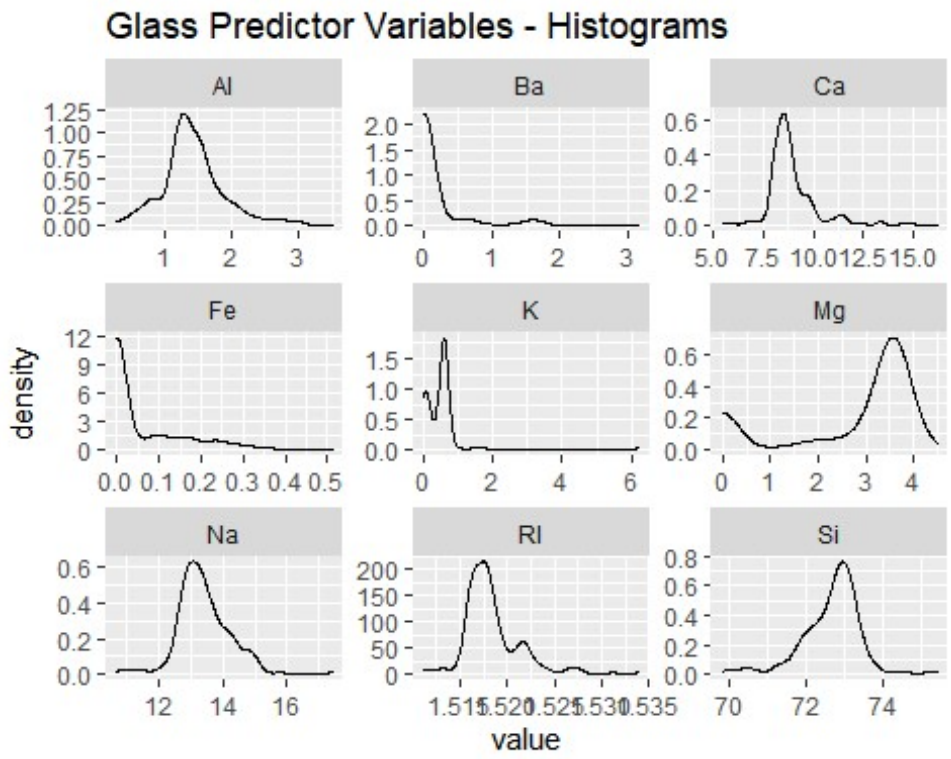
The UC Irvine Machine Learning Repository contains a data set related to glass identification. The data consist of 214 glass samples labeled as one of seven class categories. There are nine predictors, including the refractive index and percentages of eight elements: Na, Mg, Al, Si, K, Ca, Ba, and Fe.

A - Using visualizations, explore the predictor variables to understand theirdistributions as well as the relationships between predictors.

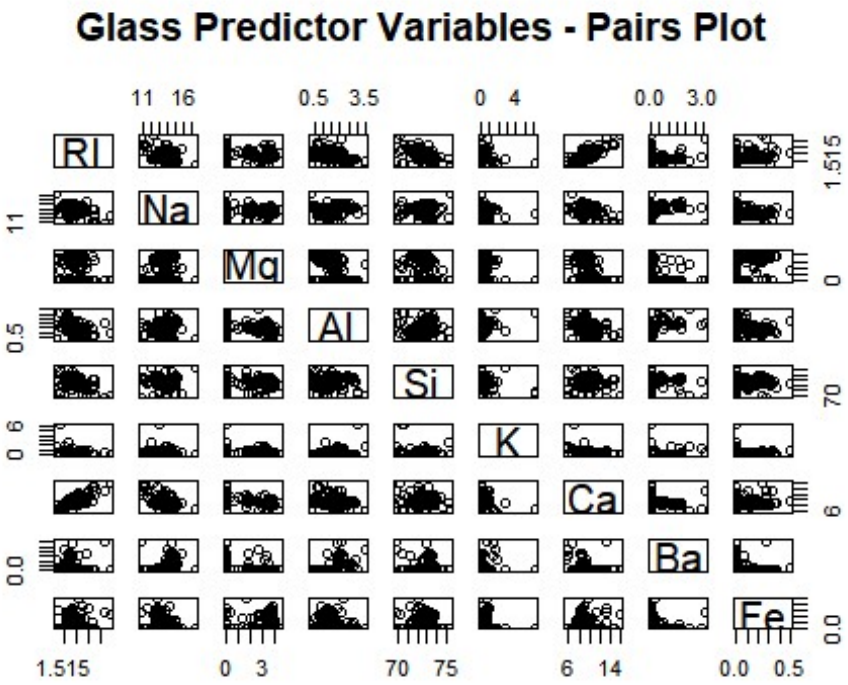
```
data(Glass)
predictors <- Glass[,1:9]
predictors %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram()+
    ggtitle("Glass Predictor Variables - Histograms")
```



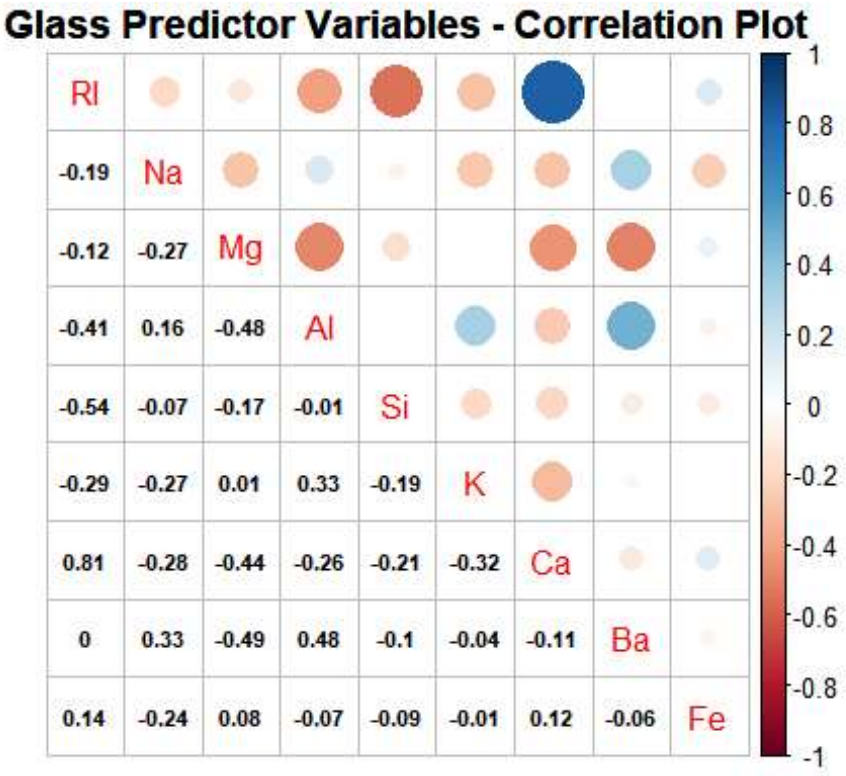

```
predictors %>%
  gather() %>%
  ggplot(aes(value)) +
  geom_density() +
  facet_wrap(~key, scales = 'free')+
  ggtitle("Glass Predictor Variables - Histograms")
```




```
pairs(predictors, main="Glass Predictor Variables - Pairs Plot")
```



```
r <-cor(predictors)
corrplot.mixed(r,
               lower.col = "black",
               number.cex = .7,
               title="Glass Predictor Variables - Correlation Plot",
               mar=c(0,0,1,0))
```



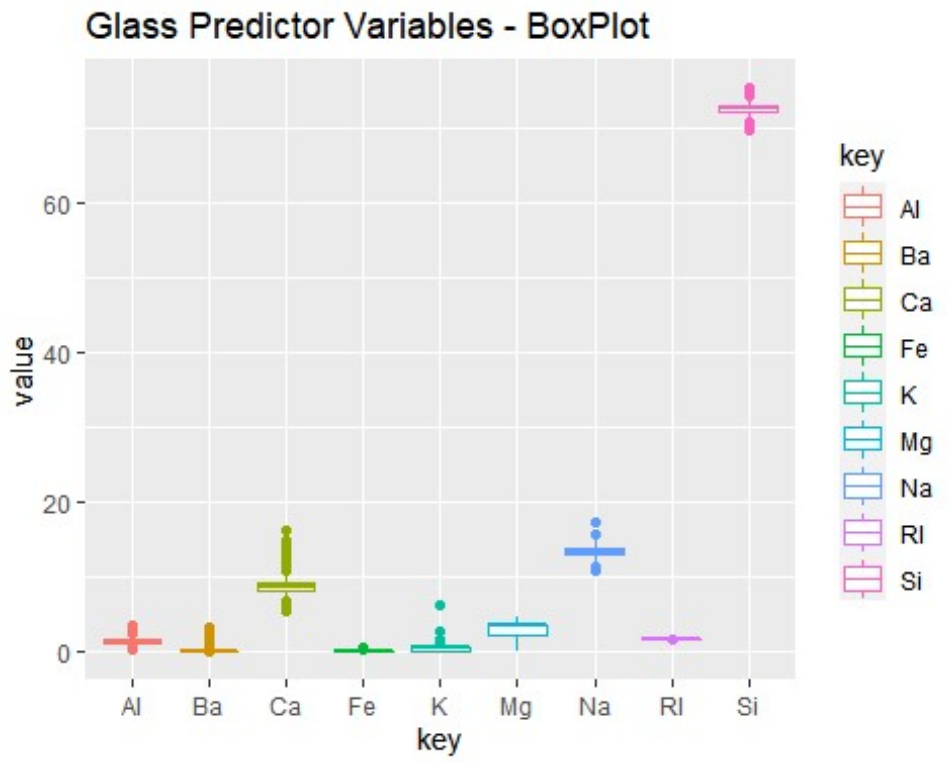
From the above plots, we can see that some of the vatiables are reasonably well centered (Al, Na), some are skewed (Mg) and there are also a few that are seem to have a high proportion of zero or near-zero weights (Fe, Ba)

Most of the predictors are negatively correlated, which makes sense. They are measuring chemical concentrations on a percentage basis. As one element increases we would expect a decrease in the others.

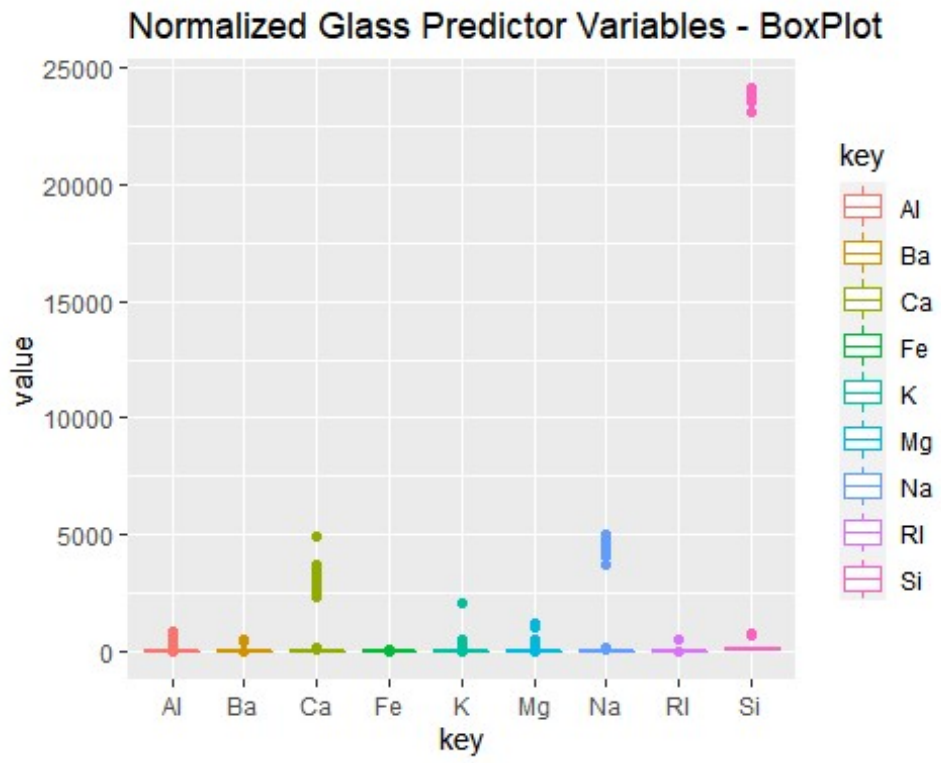
Most of the correlations are not very strong. The exception to this is the correlation between calcium oxide and the refraction index is strongly positively correlated.

B - Do there appear to be any outliers in the data? Are any predictors skewed?

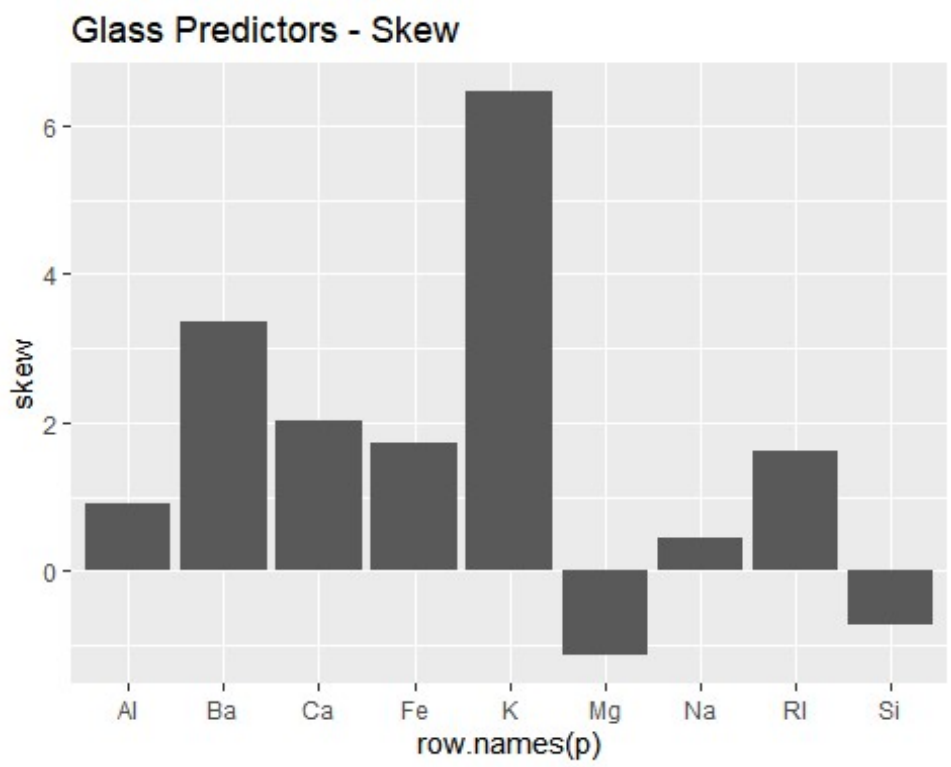
```
predictors %>%
  gather() %>%
  ggplot(aes(x=key,y=value,color=key)) +
    geom_boxplot()+
    ggtitle("Glass Predictor Variables - BoxPlot")
```




```
pred.norm <- predictors / apply(predictors, 2, sd)
pred.norm %>%
  gather() %>%
  ggplot(aes(x=key,y=value,color=key)) +
    geom_boxplot()+
    scale_y_continuous()+
    ggtitle("Normalized Glass Predictor Variables - BoxPlot")
```




```
p <-  describe(predictors)
ggplot(p,aes(x = row.names(p),y=skew))+
  geom_bar(stat='identity') +
  ggtitle("Glass Predictors - Skew")
```



In terms of the outliers, I first performed a box-plot to try to get a visual sense. I can see right away that the variables need to be re-scales. A simple/common recaling method is to divide by the min value however in this case, I have several vars with zero-mins and as such, we'll scale by the standard deviation.

Magnesium is bimodal and left skewed. Iron, potasium and barium are right skewed. The other predictors are somewhat normal.

C - Are there any relevant transformations of one or more predictors that might improve the classification model?

Something like a Box-Cox transformation might improve the classification model's preformance.

Question 3.2

The soybean data can also be found at the UC Irvine Machine Learning Repository. Data were collected to predict disease in 683 soybeans. The 35 predictors are mostly categorical and include information on the environmental conditions (e.g., temperature, precipitation) and plant conditions (e.g., left spots, mold growth). The outcome labels consist of 19 distinct classes.

A - Investigate the frequency distributions for the categorical predictors. Are any of the distributions degenerate in the ways discussed earlier in this chapter?

```
data(Soybean)
#number of unique values per col
incl.nas <- sapply(sapply(Soybean,unique),length)
no.nas <- sapply(sapply(Soybean[complete.cases(Soybean),],unique),length)

r <- t(rbind(incl.nas,no.nas))
row.names(r) <- colnames(Soybean)
kable(r)
```

	incl.nas	no.nas
Class	19	15
date	8	7
plant.stand	3	2
precip	4	3
temp	4	3
hail	3	2
crop.hist	5	4
area.dam	5	4
sever	4	3
seed.tmt	4	3
germ	4	3
plant.growth	3	2
leaves	2	2
leaf.halo	4	3
leaf.marg	4	3
leaf.size	4	3
leaf.shread	3	2
leaf.malf	3	2
leaf.mild	4	3
stem	3	2
lodging	3	2
stem.cankers	5	4
canker.lesion	5	4
fruiting.bodies	3	2
ext.decay	4	2
mycelium	3	2
int.discolor	4	3

sclerotia	3	2
fruit.pods	5	3
fruit.spots	5	4
seed	3	2
mold.growth	3	2
seed.discolor	3	2
seed.size	3	2
shriveling	3	2
roots	4	3

The table above shows the unique-value-count by variable. Based on this table it does not appear as though there are any variables with degenerate distributions.

B - Roughly 18% of the data are missing. Are there particular predictors that are more likely to be missing? Is the pattern of missing data related to the classes?

```
Soybean.incomplete <- Soybean[!complete.cases(Soybean),]
missing.cols <- Soybean.incomplete %>%
  select(everything()) %>%  # replace to your needs
  summarise_all(funs(sum(is.na(.))))
missing.cols <- t(missing.cols/nrow(Soybean))
missing.cols <- missing.cols[order(-missing.cols),]
kable(missing.cols)
```

	x
hail	0.1771596
sever	0.1771596
seed.tmt	0.1771596
lodging	0.1771596
germ	0.1639824
leaf.mild	0.1581259
fruiting.bodies	0.1551977
fruit.spots	0.1551977
seed.discolor	0.1551977
shriveling	0.1551977
leaf.shread	0.1464129
seed	0.1346999
mold.growth	0.1346999
seed.size	0.1346999
leaf.halo	0.1229868
leaf.marg	0.1229868

leaf.size	0.1229868
leaf.malf	0.1229868
fruit.pods	0.1229868
precip	0.0556369
stem.cankers	0.0556369
canker.lesion	0.0556369
ext.decay	0.0556369
mycelium	0.0556369
int.discolor	0.0556369
sclerotia	0.0556369
plant.stand	0.0527086
roots	0.0453880
temp	0.0439239
crop.hist	0.0234261
plant.growth	0.0234261
stem	0.0234261
date	0.0014641
area.dam	0.0014641
Class	0.0000000
leaves	0.0000000

```
case.count <- Soybean.incomplete %>%
  group_by(Class) %>%
  tally()
na.count <- aggregate(Soybean.incomplete, list(Soybean.incomplete$Class),
function(x) sum(is.na(x)))
case.count$NAs <- data.frame(rowSums(na.count[2:ncol(na.count)]))
colnames(case.count) <- c("Class","Incomeplete.Cases", "NA.Values")
case.count$NA.Per.Case <- case.count$NA.Values / case.count$Incomeplete.Cases
kable(case.count)
```

Class	Incomeplete.Cases	NA.Values	NA.Per.Case
2-4-d-injury	16	450	28.12500
cyst-nematode	14	336	24.00000
diaporthe-pod-&-stem-blight	15	177	11.80000
herbicide-injury	8	160	20.00000
phytophthora-rot	68	1214	17.85294

Variables

There does seem to be a pattern in some of the variables which are missing.

- crop damage (hail, lodging, severe weather) appear to be among the most common missing variables (~18%)
- next most common are various seed & fruit related metrics

C - Develop a strategy for handling missing data, either by eliminating predictors or imputation.

For this kind of problem, I;d liek to try a "one size fits all" solution is rarely optimal.

Rare Exogenous Events - Impute Zeros

There are several variables where I feel imputation makes no sense - For these variables, we'll assume an NA means that they didn't occur and impute zeros

```
Soybean$hail[is.na(Soybean$hail)] <- 0
Soybean$sever[is.na(Soybean$hail)] <- 0
```

Remaining Data - Knn Impute

For the remaining data we'll use KNN (k=10) to impute. Note that I'm using the mode rather than an average as all of these variables appear to be discreet.

```
df <- data.frame(Soybean)
Soybean.impute <- knnImputation(df, k = 10, scale = T, meth = "mode",
               distData = NULL)
nrow(Soybean.impute[!complete.cases(Soybean.impute),])

## [1] 0
```

I can see that the number of incomplete cases is now 0.