Week 5 Assignment

Chi Hang(Philip) Cheung

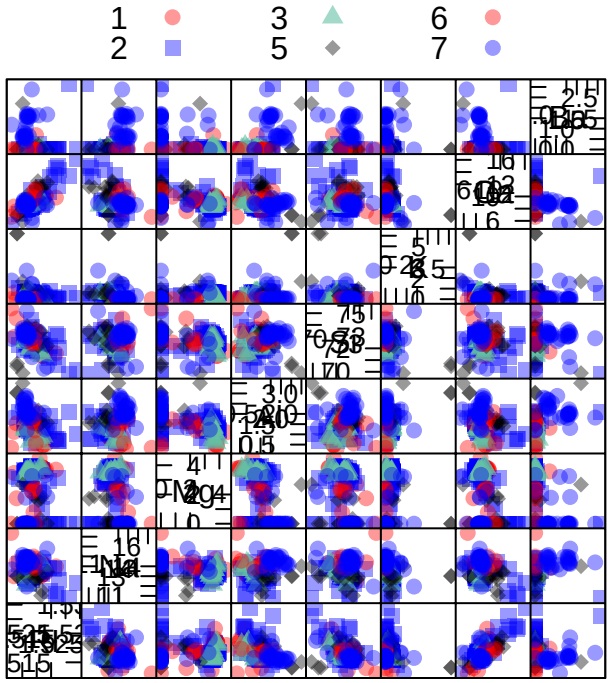
2025-03-01

```
library(AppliedPredictiveModeling)
library(mlbench)
library(caret)
library(ggplot2)
library(tidyverse)
```

3.1 a)

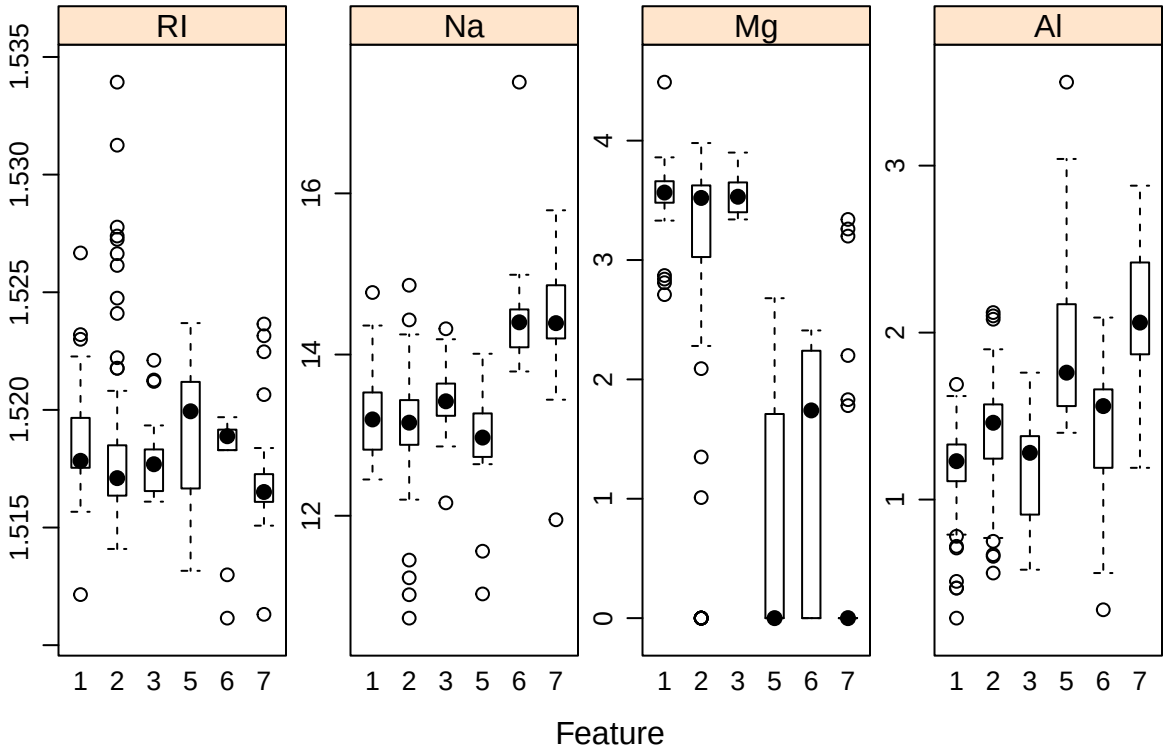
Each predictor are compared against each other in on plot.

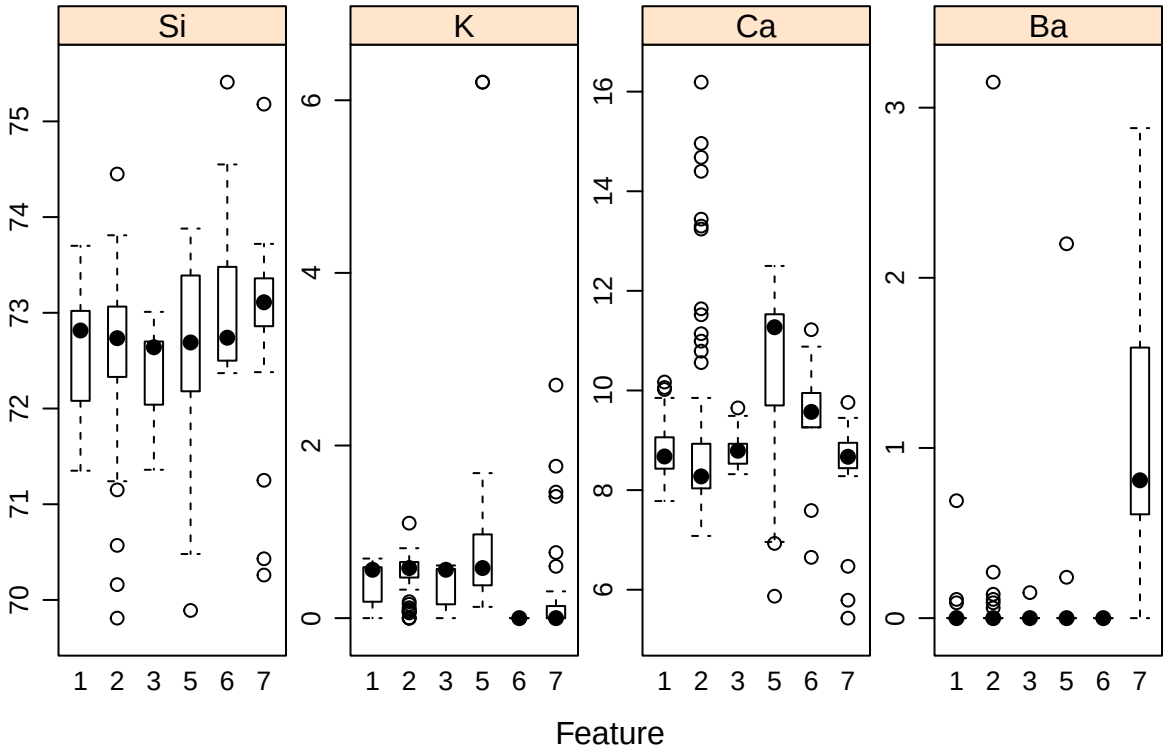
```
data(Glass)
#creating a feature plot to see the relationships between each element and their respective Types:
transparentTheme(trans = 0.4)
featurePlot(x = Glass[, 1:8],
            y = Glass$Type,
            plot = "pairs",
            auto.key = list(columns = 3,
                            title = 'Glass elements and relationships'))
```



Scatter Plot Matrix

```
featurePlot(x = Glass[, 1:8],
            y = Glass$Type,
            plot = "box",
            scales = list(y = list(relation='free')),
            layout = c(4,1),
            auto.key = list(title = 'Box Plot for each elements and the RI based on glass types'))
```


RI
Na
Mg
Al
1.515
1.520
1.525
1.530
1.535
12
14
16
0
1
2
3
4
1
2
3
5
6
7
Feature

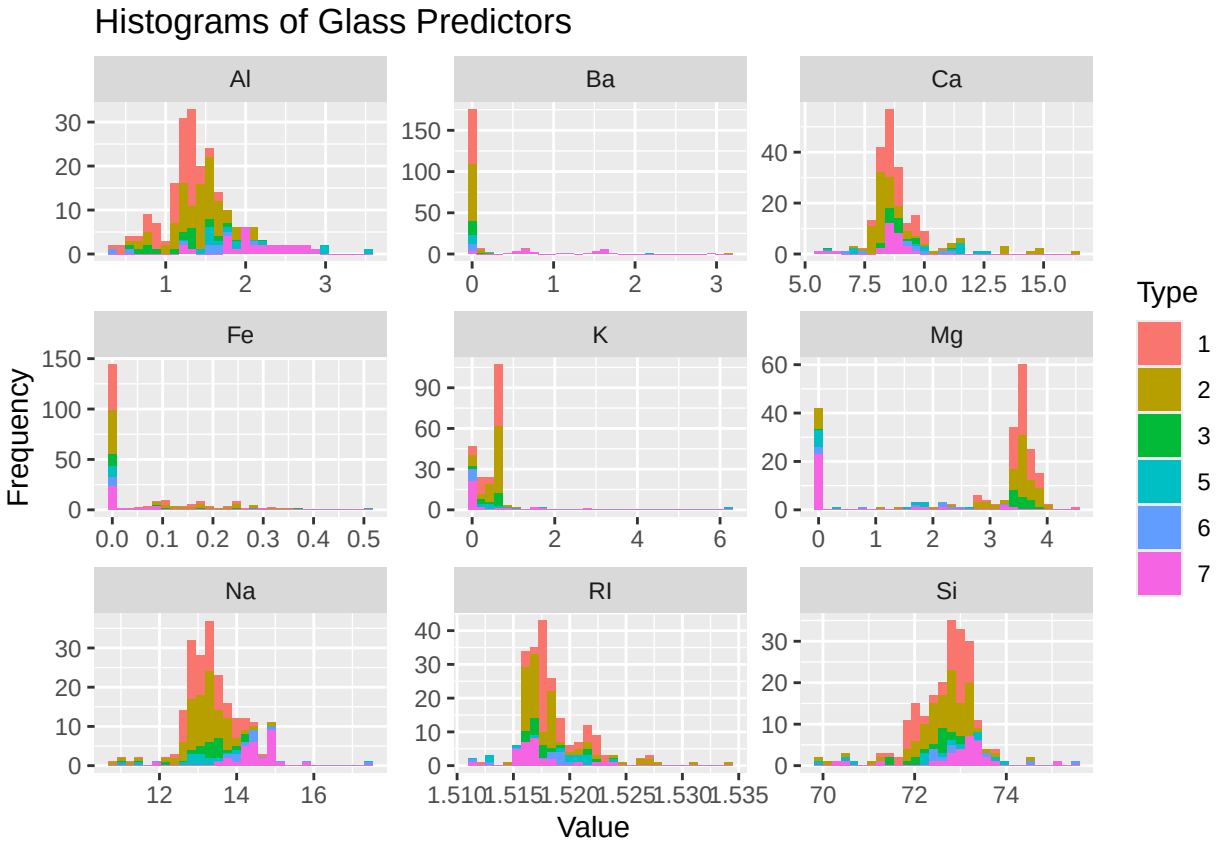


(b)Do the reappear to be any outliers in the data? Are any predictors skewed?

Ans: There are outliers as indicated in the box plot above. Almost each type of the glass has outliers in all categories. As for skewedness, Ba, Fe, and K all demonstrated right skewedness. K has a left skewedness. RI, Ca, Al, and Na have a slight right skewedness.

```
long_glass<- Glass %>%
  pivot_longer(cols = 1:9,
               names_to = 'predicators',
               values_to = 'values')
OG <- long_glass %>%
  ggplot(aes(x=values, fill = Type)) +
  geom_histogram()+
  facet_wrap(~ predicators, scale = 'free')+
  labs(title = "Histograms of Glass Predictors", x = "Value", y = "Frequency")
OG
```

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



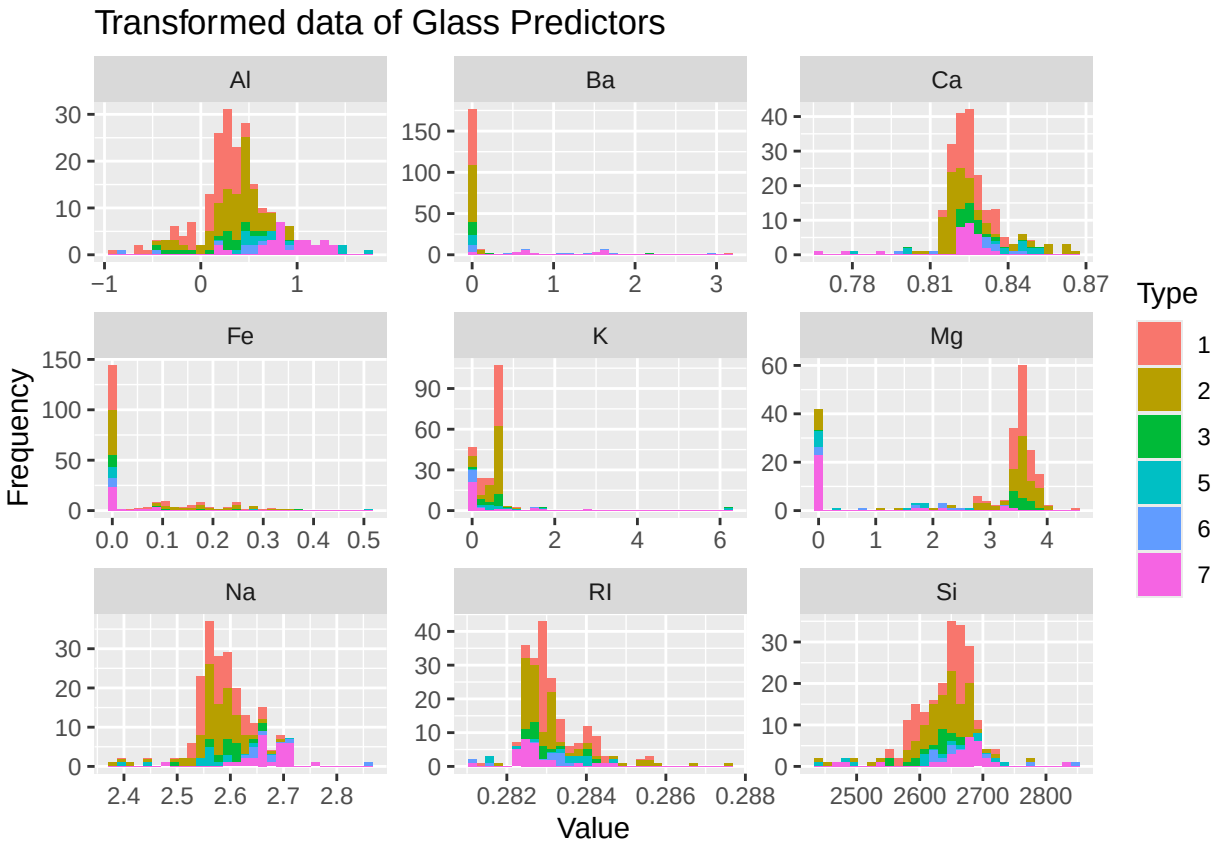
(c)Are there any relevant transformations of one or more predictors that might improve the classification model?

Ans: I applied the Box Cox transformation to all the predicators in the Glass data set. There are some improvements to centering the distributions for Al, Ca, and Na. Other predicators such as Ba, Fe, K, and Mg have no effect. I believe this is because those predicators have large amount of 0 value within the set, rendering any transformation useless due to zero value.

```
#caret package box-cox transformation:
#to select only the predictors, except the type:
glass_data = Glass[, 1:9]

#use sapply to apply transformation on all columns:
glass_bc<- sapply(glass_data, function(x) predict(BoxCoxTrans(x), x))

#Put the transformed data back into dataframe:
glass_bc<- as.data.frame(glass_bc)
#re-combine with the types:
glass_bc<- cbind(glass_bc, Glass$Type)
names(glass_bc)[names(glass_bc)=="Glass$Type"]<-'Type'

glass_bc_long<- glass_bc %>%
  pivot_longer(cols = 1:9,
               names_to = 'predicators',
               values_to = 'values')
glass_bc_long %>%
  ggplot(aes(x=values, fill = Type)) +
```

```
  geom_histogram()+
  facet_wrap(~ predicators, scale = 'free')+
  labs(title = "Transformed data of Glass Predictors", x = "Value", y = "Frequency")
```

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



3.2 Soybean dataset

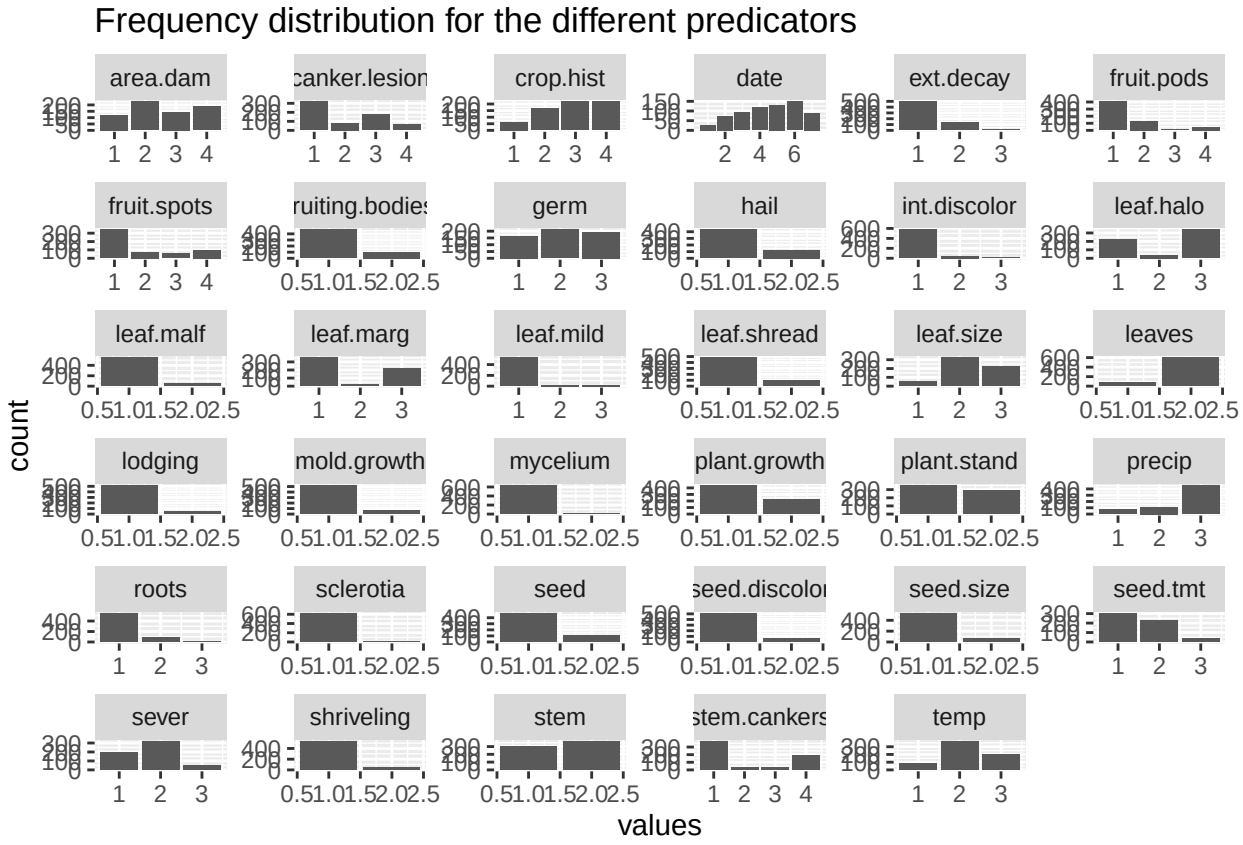
a) Investigate the frequency distributions for the categorical predictors. Are any of the distributions degenerate in the ways discussed earlier in this chapter?

Ans: The potential degenerates are mycelium, leaf mild, sclerotia, and leaves. These ones have almost all one distinct value.

```
data("Soybean")

soyb_long<- Soybean %>%
  select(c(-1)) %>%
  mutate(across(everything(), as.numeric)) %>%
  pivot_longer(cols=everything(),
               names_to = 'predicators',
               values_to = 'values'
               )
soyb_long %>%
```

```
  ggplot(aes(x=values)) +
  geom_bar()+
  facet_wrap(~ predicators, scale='free') +
  labs(title='Frequency distribution for the different predicators')
```

```
## Warning: Removed 2337 rows containing non-finite outside the scale range
## (‘stat_count()‘).
```



(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?

Ans:The ones with the highest NA data rae hail, lodging, seed.tmt, and sever. There seems to be no pattern related to the missing data.

```
soyb_long %>%
  group_by(predicators, values) %>% #group by the two variables
  count() %>%
  group_by(predicators) %>%
  mutate(percentage = n / sum(n)*100) %>% #calculated the percentage of NA values
  ungroup() %>%
  filter(is.na(values)) %>%
  arrange(desc(percentage)) #re-arrange to descending orders
```

```
## # A tibble: 34 x 4
```

```
##    predicators     values     n percentage
##    <chr>            <dbl> <int>      <dbl>
##  1 hail                NA   121       17.7
##  2 lodging             NA   121       17.7
##  3 seed.tmt            NA   121       17.7
##  4 sever               NA   121       17.7
##  5 germ                NA   112       16.4
##  6 leaf.mild           NA   108       15.8
##  7 fruit.spots         NA   106       15.5
##  8 fruiting.bodies     NA   106       15.5
##  9 seed.discolor       NA   106       15.5
## 10 shriveling          NA   106       15.5
## # i 24 more rows
```

c) Develop a strategy for handling missing data, either by eliminating predictors or imputation.

Ans: For the soybean data, the highest percentage for missing data for a single predicator is about 17.7%. About 82.3% of the data are still intact. There is no reason to eliminate the predicator altogether due to less than 50% data loss. I would prefer to use imputation to attempt estimating the missing data, possibly by the columns' mean value. However, this method is also prone to errors due to how the data structure is in the Soybean data. Many of these predicators are binaries and 0-3 values. Imputation will always omit the 0 value due to how means are calculated. Nevertheless, data elimination is to be avoided if the missing values are not as significant.