HW1

Prutha

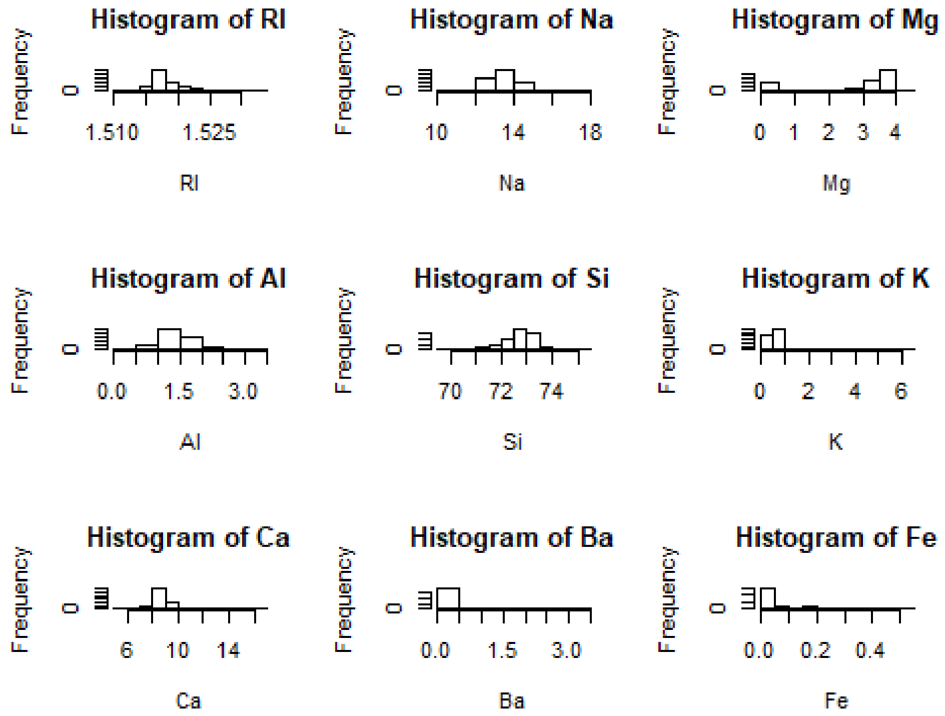
September 24, 2019

Question 3.1

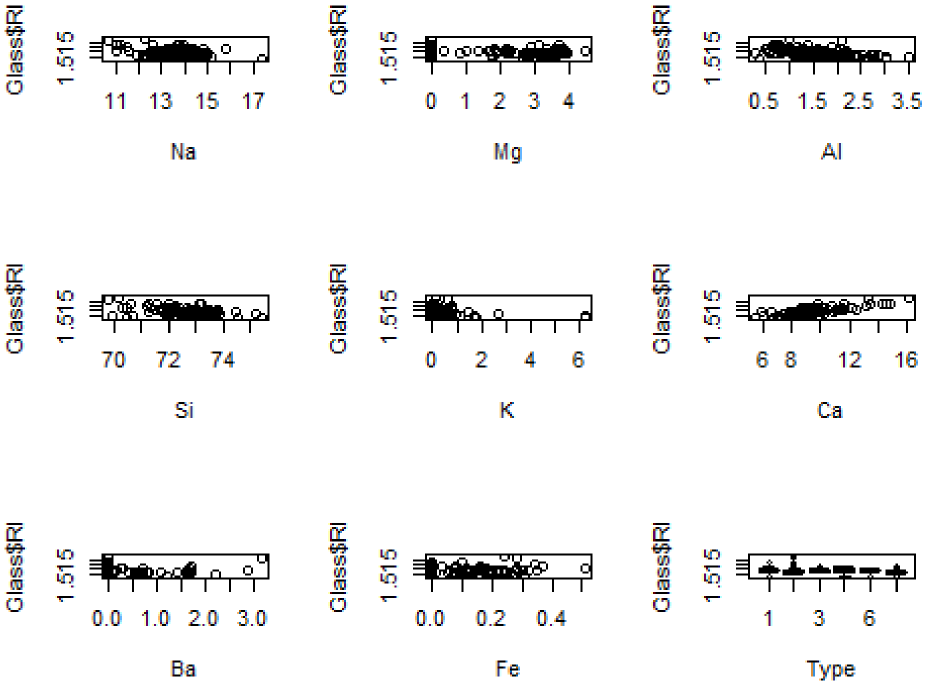
(a)Using visualizations, explore the predictor variables to understand their distributions as well as the relationships between predictors.

Answer: Below are the histograms displaying the frequency of each predictor, which helps us understand their distribution better. Also, we have created the plots of every predictor with respect to every other predictor in order to understand the relation in between them. We observe all the predictors are moderately skewed expect for a few predictors such as K, Ba and Fe.

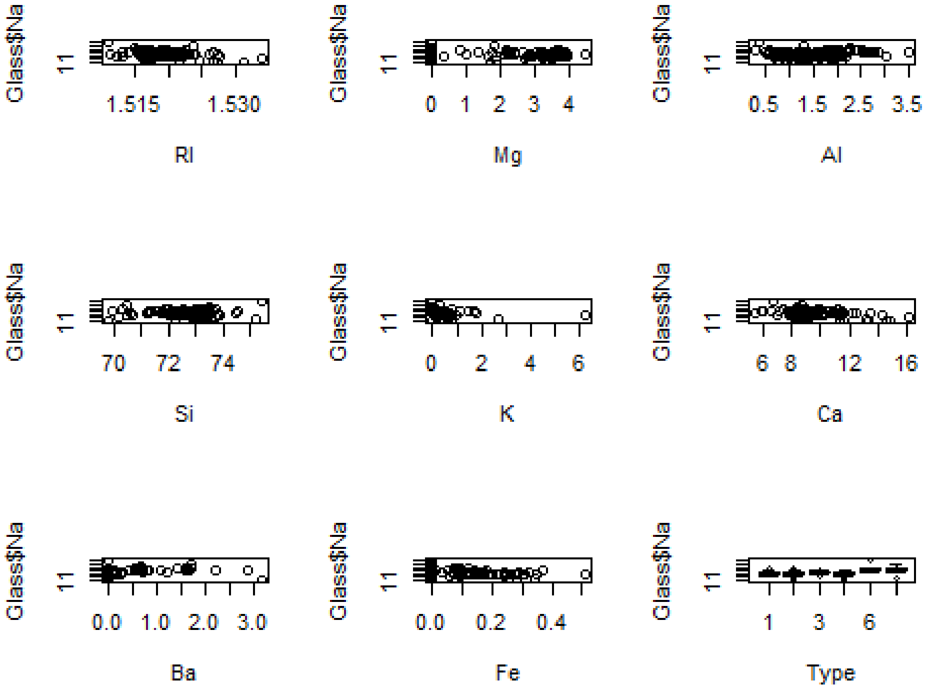
```
#3.1 (a)
```



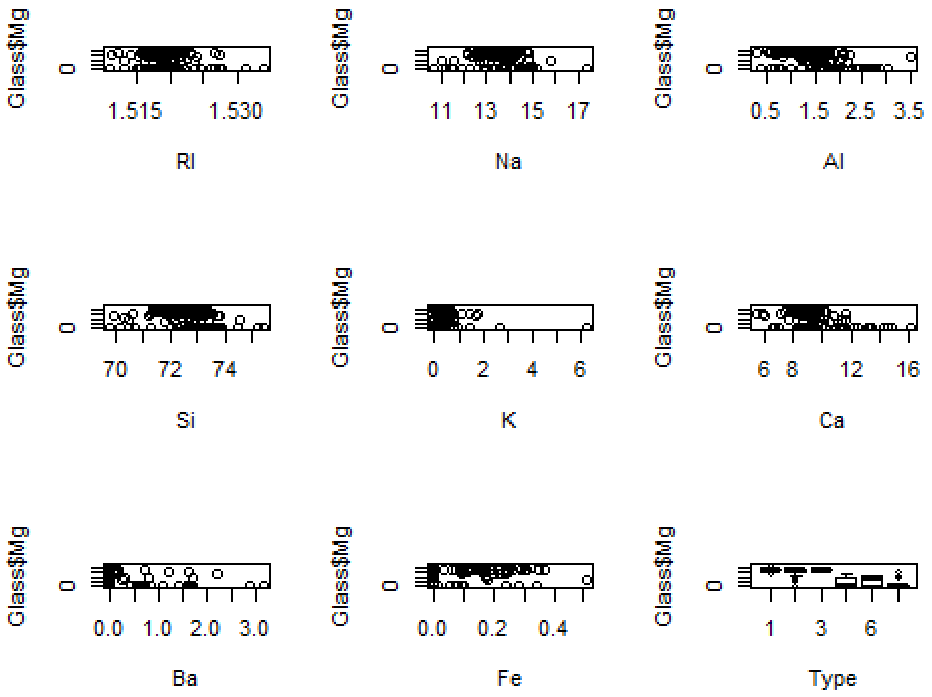
```
par(mfrow = c(3,3))
plot(Glass$RI~.,data = Glass)
```



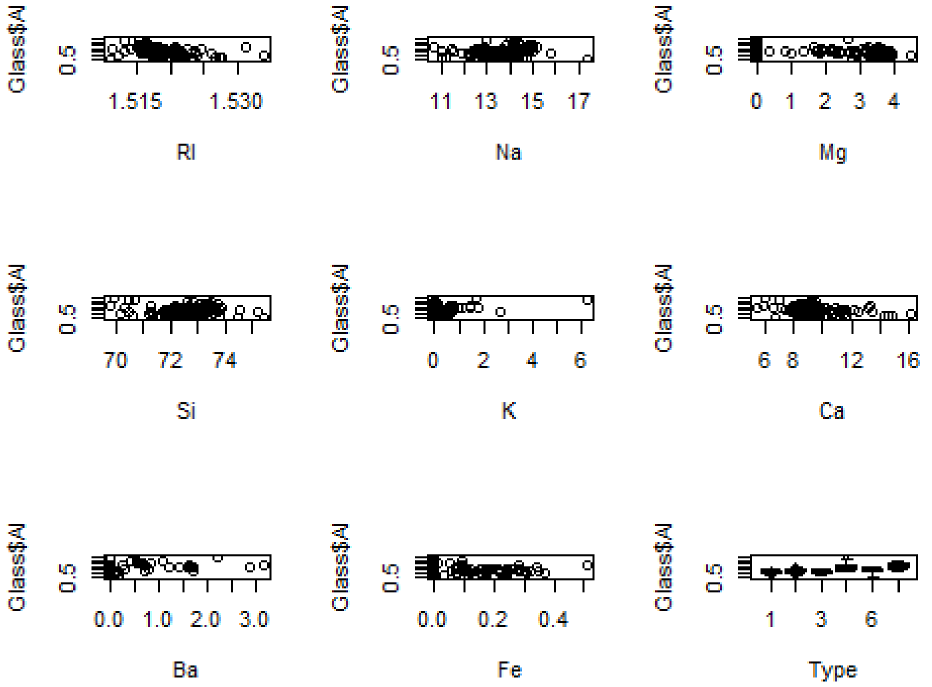
```
plot(Glass$Na~.,data = Glass)
```



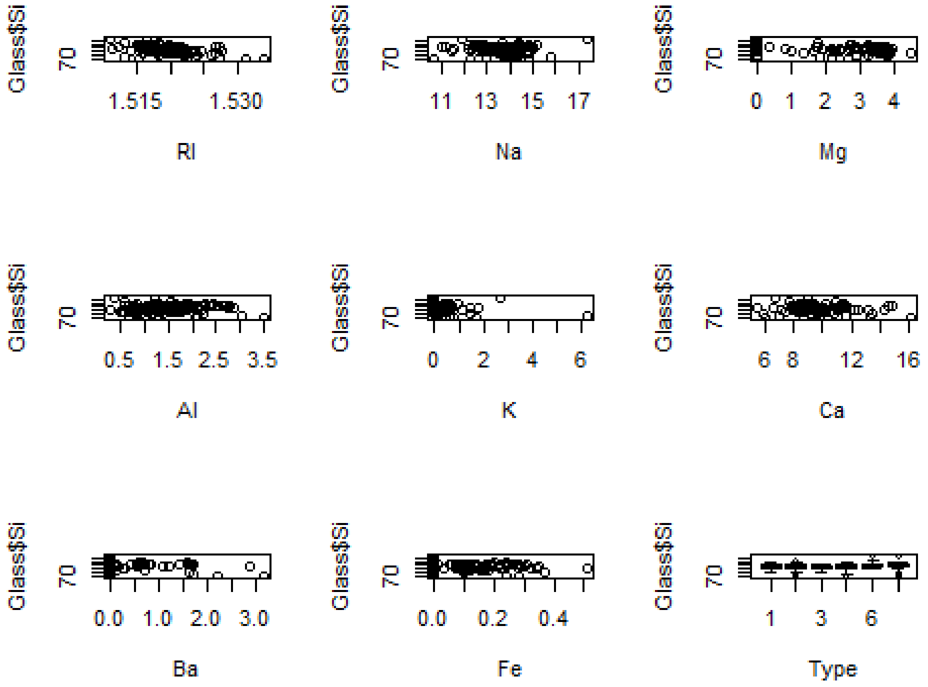
```
plot(Glass$Mg~.,data = Glass)
```



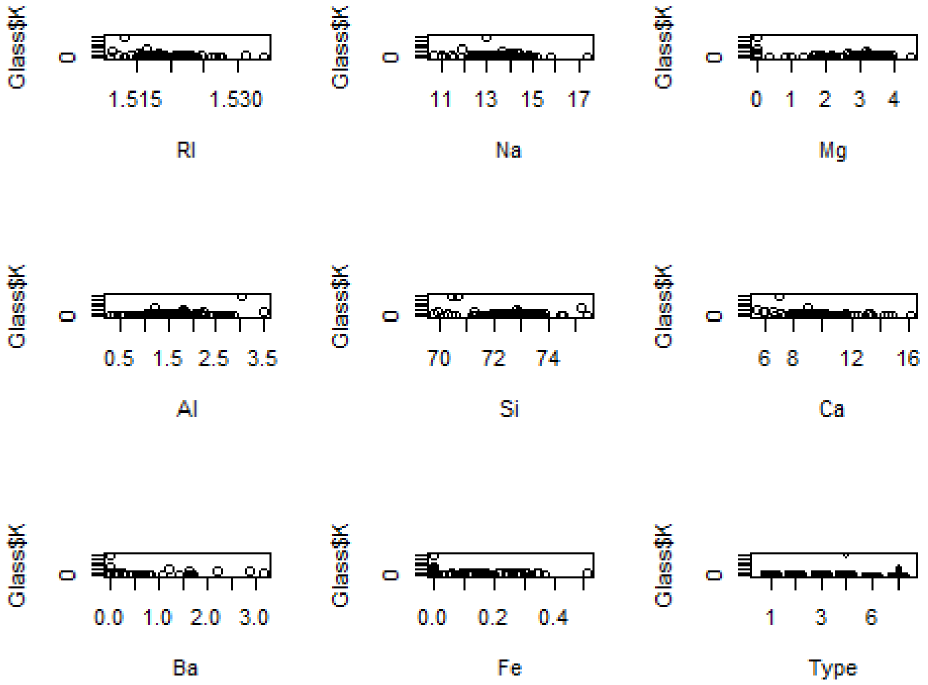
```
plot(Glass$Al~.,data = Glass)
```



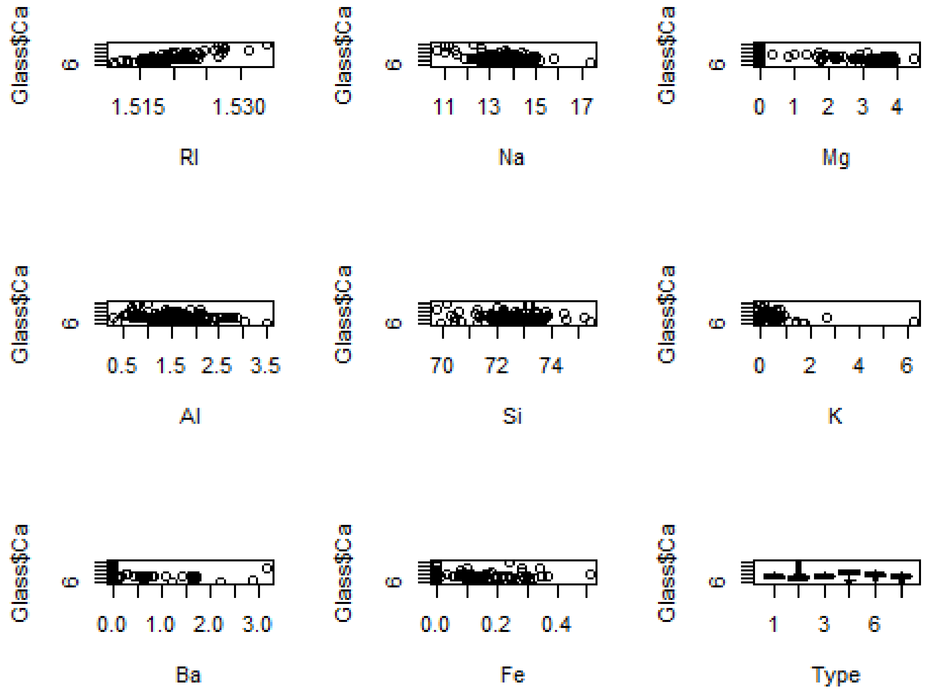
```
plot(Glass$Si~.,data = Glass)
```



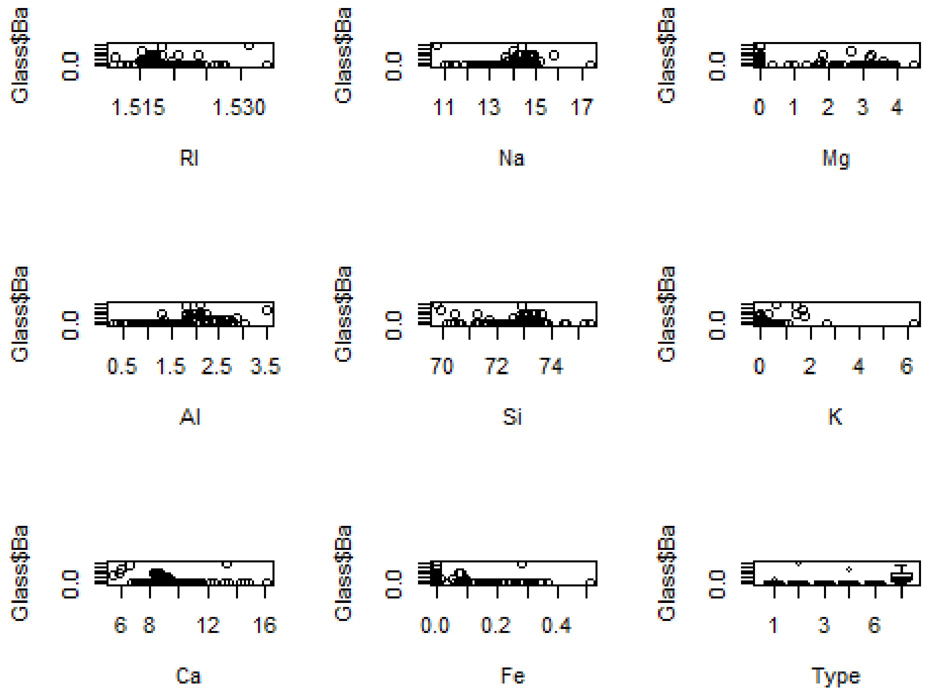
```
plot(Glass$K~.,data = Glass)
```



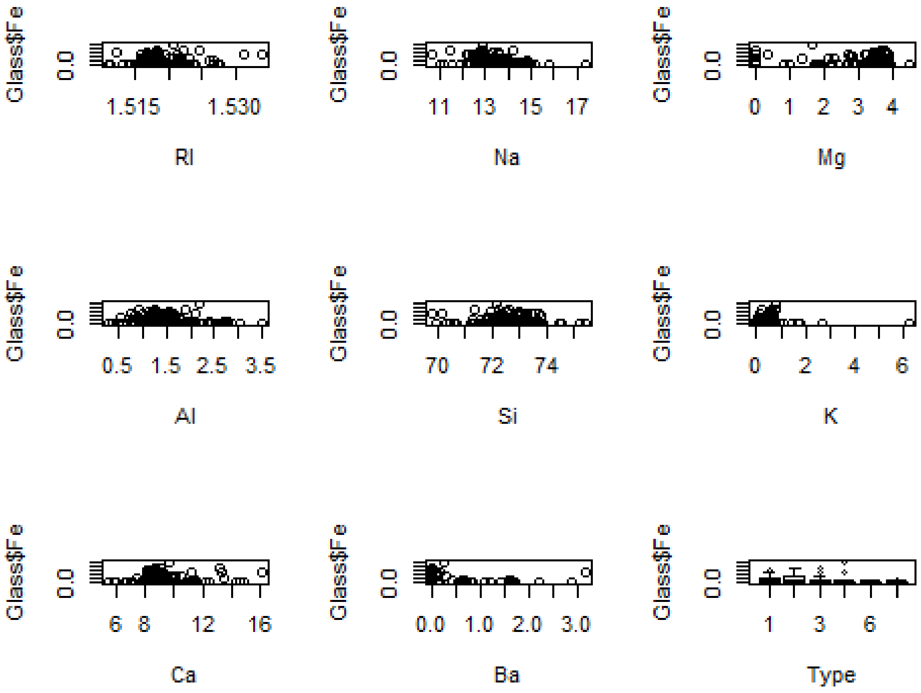
```
plot(Glass$Ca~.,data = Glass)
```



```
plot(Glass$Ba~.,data = Glass)
```

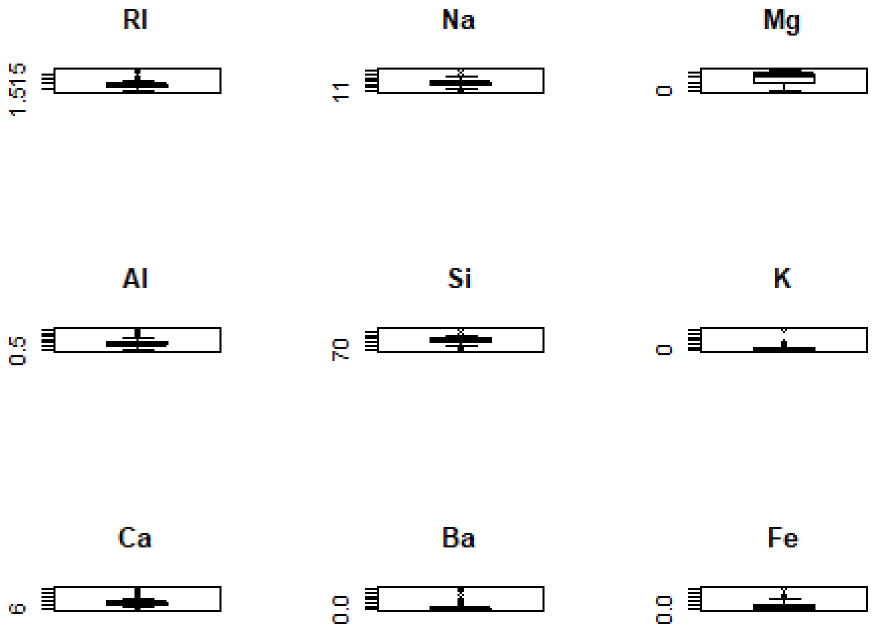


```
plot(Glass$Fe~.,data = Glass)
```


Glass$Fe
0.0
1.515
1.530
RI
Glass$Fe
0.0
11
13
15
17
Na
Glass$Fe
0.0
0
1
2
3
4
Mg
Glass$Fe
0.0
0.5
1.5
2.5
3.5
Al
Glass$Fe
0.0
70
72
74
Si
Glass$Fe
0.0
0
2
4
6
K
Glass$Fe
0.0
6
8
12
16
Ca
Glass$Fe
0.0
0.0
1.0
2.0
3.0
Ba
Glass$Fe
0.0
1
3
6
Type

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

Answer: Yes, all the predictors have some outliers in their data, expect for the predictor ‘Mg’. We observe all the predictors are moderately skewed expect for a few predictors such as **K**, **Ba** and **Fe**.



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

Answer: we apply transformation method called as ‘**Box Cox Transformation**’. After applying this transformation method, we observe that the data is somewhat moderately skewed.

```
# (c) Transformation

#Box Cox tranformtion
library(caret)

xx1 <- preProcess(Glass, method = c("BoxCox"))
xx1

## Created from 214 samples and 6 variables
##
```

```
## Pre-processing:
##   - Box-Cox transformation (5)
##   - ignored (1)
##
## Lambda estimates for Box-Cox transformation:
## -2, -0.1, 0.5, 2, -1.1
```

```
# Apply the transformations:

transformed <- predict(xx1, Glass)
transformed
```

```
skewness(transformed$RI)
```

```
## [1] 1.56566
```

```
skewness(transformed$Na)
```

```
## [1] 0.03384644
```

```
skewness(transformed$Mg)
```

```
## [1] -1.136452
```

```
skewness(transformed$Al)
```

```
## [1] 0.09105899
```

```
skewness(transformed$Si)
```

```
## [1] -0.6509057
```

```
skewness(transformed$K)
```

```
## [1] 6.460089
```

```
skewness(transformed$Ca)
```

```
## [1] -0.1939557
```

```
skewness(transformed$Ba)
```

```
## [1] 3.36868
```

```
skewness(transformed$Fe)
```

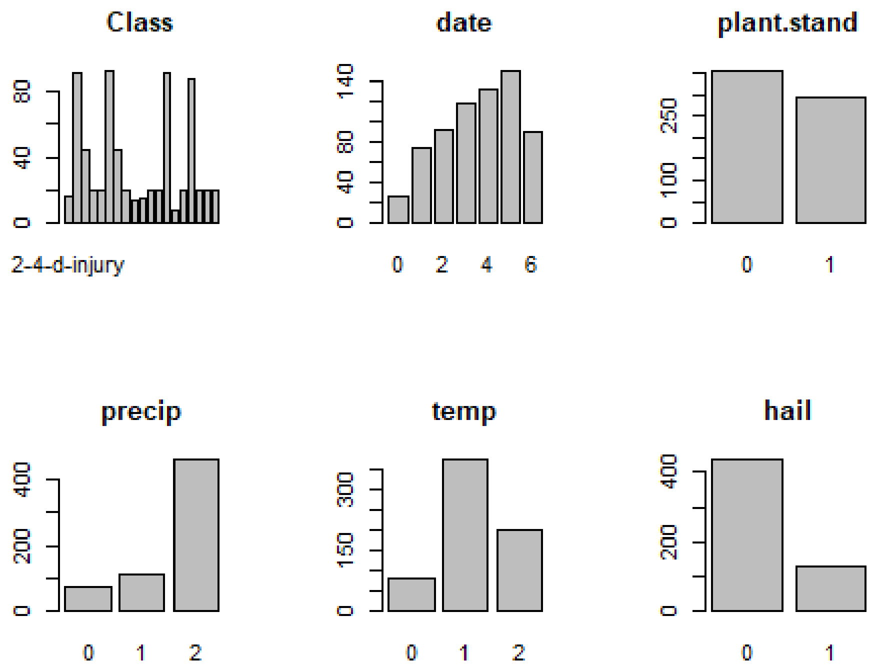
```
## [1] 1.729811
```

Question 3.2

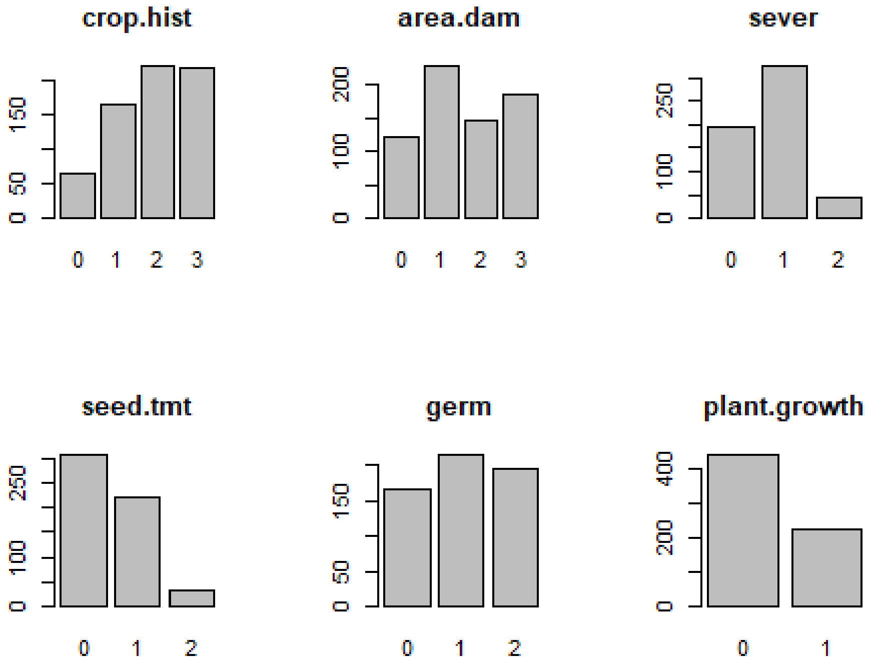
(a) Investigate the frequency distributions for the categorical predictors. Are any of the distributions degenerate in the ways discussed earlier in this chapter?

Answer: Below are the plots displaying the frequency of each predictor, which helps us understand their distribution better. Predictors whose variance is zero are the ones of degenrated distribution which Leaf Mild, Mycelium and Sclerotia

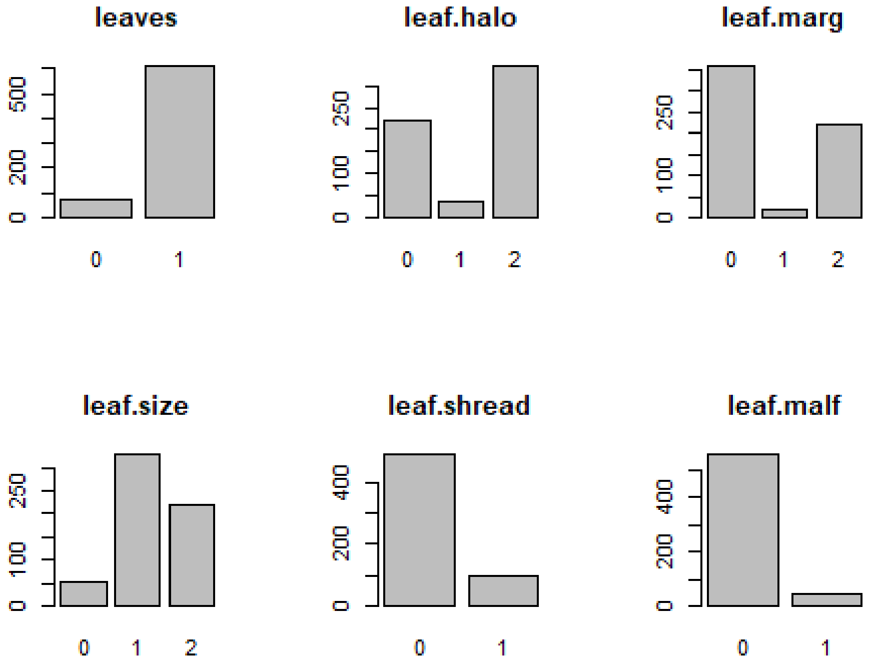
```
# Q3.2
#frequency distributions for the categorical predictors

data(Soybean)
par(mfrow = c(2,3))

plot(Soybean$Class, main = "Class")
plot(Soybean$date, main = "date")
plot(Soybean$plant.stand, main = "plant.stand")
plot(Soybean$precip, main = "precip")
plot(Soybean$temp, main = "temp")
plot(Soybean$hail, main = "hail")
```



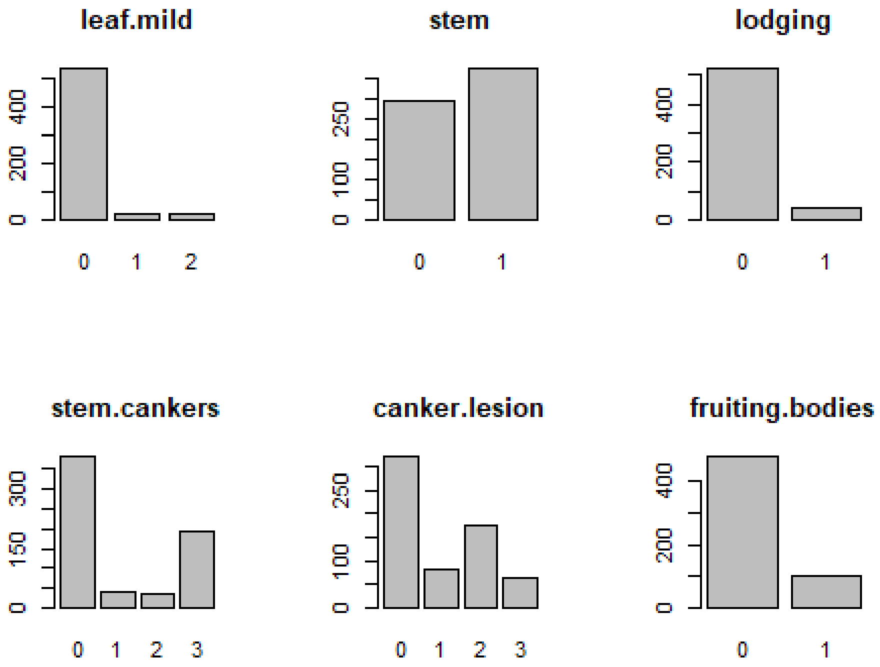
```
par(mfrow = c(2,3))

plot(Soybean$crop.hist, main = "crop.hist")
plot(Soybean$area.dam, main = "area.dam")
plot(Soybean$sever, main = "sever")
plot(Soybean$seed.tmt, main = "seed.tmt")
plot(Soybean$germ, main = "germ")
plot(Soybean$plant.growth, main = "plant.growth")
```



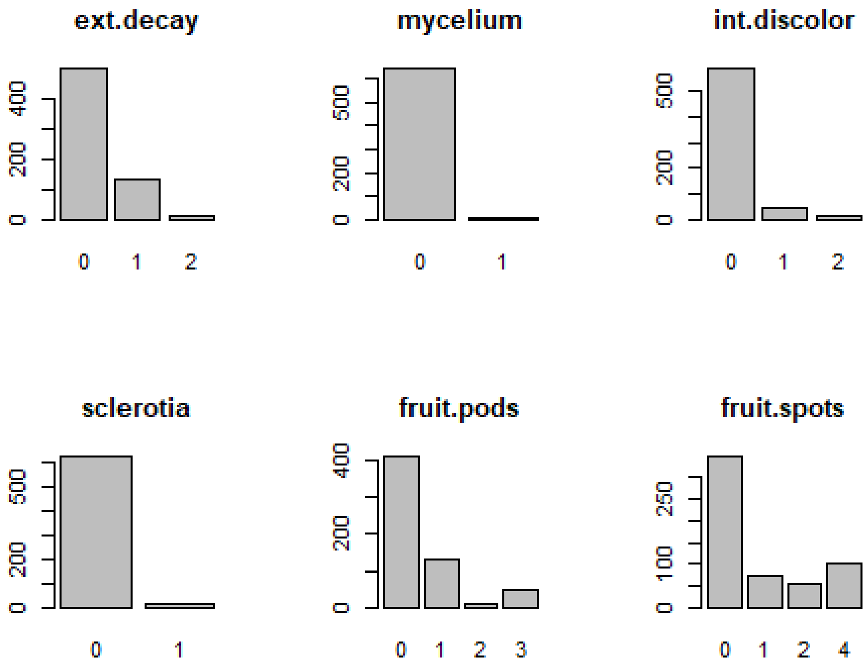
```
par(mfrow = c(2,3))

plot(Soybean$leaves, main = "leaves")
plot(Soybean$leaf.halo, main = "leaf.halo")
plot(Soybean$leaf.marg, main = "leaf.marg")
plot(Soybean$leaf.size, main = "leaf.size")
plot(Soybean$leaf.shread, main = "leaf.shread")
plot(Soybean$leaf.malf, main = "leaf.malf")
```



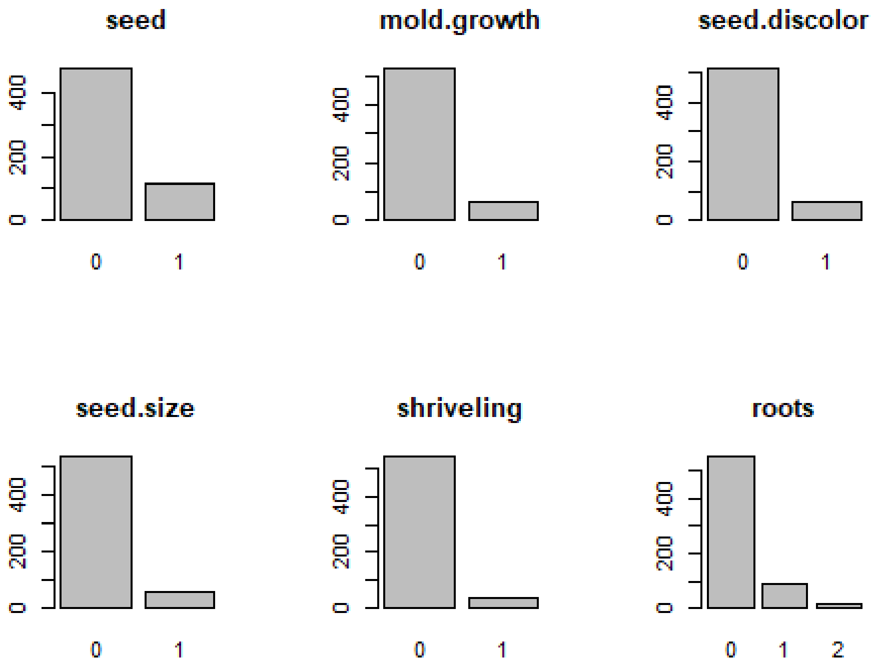
```r
par(mfrow = c(2,3))

plot(Soybean$leaf.mild, main = "leaf.mild")
plot(Soybean$stem, main = "stem")
plot(Soybean$lodging, main = "lodging")
plot(Soybean$stem.cankers, main = "stem.cankers")
plot(Soybean$canker.lesion, main = "canker.lesion")
plot(Soybean$fruiting.bodies, main = "fruiting.bodies")
```



```r
par(mfrow = c(2,3))

plot(Soybean$ext.decay, main = "ext.decay")
plot(Soybean$mycelium, main = "mycelium")
plot(Soybean$int.discolor, main = "int.discolor")
plot(Soybean$sclerotia, main = "sclerotia")
plot(Soybean$fruit.pods, main = "fruit.pods")
plot(Soybean$fruit.spots, main = "fruit.spots")
```



```r
par(mfrow = c(2,3))

plot(Soybean$seed, main = "seed")
plot(Soybean$mold.growth, main = "mold.growth")
plot(Soybean$seed.discolor, main = "seed.discolor")
plot(Soybean$seed.size, main = "seed.size")
plot(Soybean$shriveling, main = "shriveling")
plot(Soybean$roots, main = "roots")
```



(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?

Answer: We divide the data into subsets for convience and then check the NA values present in data. We observe that most of the missing data seems to come from columns such as “**Seed.tmt, Sever, Germ, Leaf Halo, Hail, Leaf Mild, Fruiting Bodies, Leaf Marg, Leaf size, Leaf shred, Leaf Maf, Fruiting Bodies, Lodging, Fruit pods, fruits spots, mold growth, Shriveling and Seed Discolor**.” All of these predictors have over more than 80 missing values. Also, the missing data is related to the classes, as after sub seting we observe that the missing values are dependent on the class. After adding the na values, only specific classes have those values while the rest classes do not appear to have the missing values.

```
# to calculate variance

nearZeroVar(Soybean)

## [1] 19 26 28

#3.2 (b)
# Diving the data into subset
```

(c) Develop a strategy for handling missing data, either by eliminating

predictors or imputation.

Answer: To handle the missing data, we first try eliminating the na values however it does not work quite efficiently.

Hence, we try the second method of Imputation, however missing values are not removed completely which brings us to a conclusion that both of these methods are not working well to deal/remove the missing values

Question 3.3

(a) Load the data

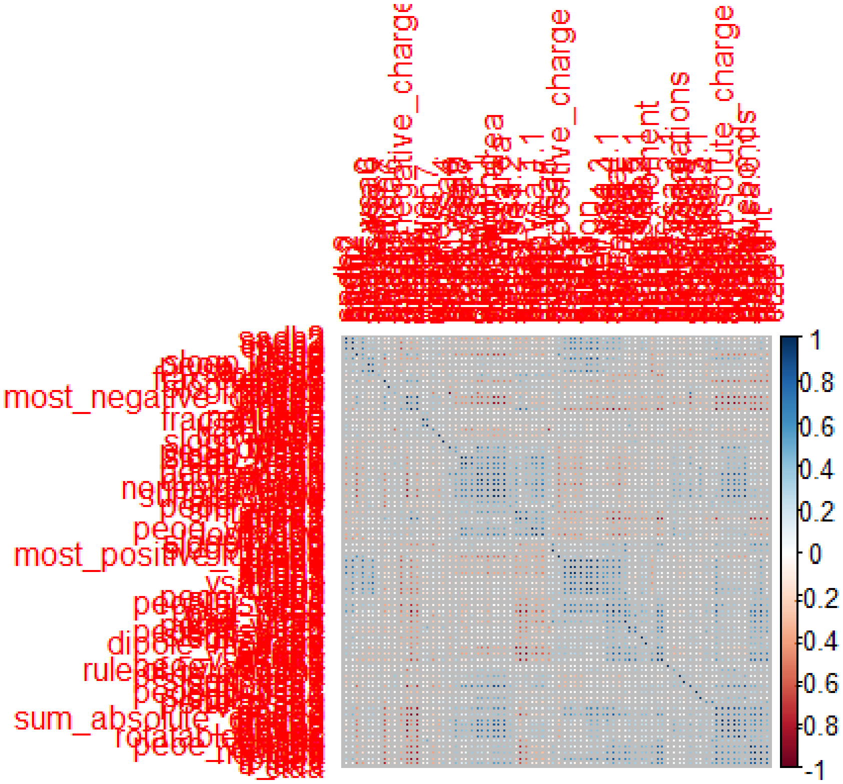
```
# 3.3 (a)
library(caret)
library(corrplot)

## Warning: package 'corrplot' was built under R version 3.5.3

## corrplot 0.84 loaded
```

```
data(BloodBrain)
```

(c) Generally speaking, are there strong relationships between the predictor data? If so, how could correlations in the predictor set be reduced? Does this have a dramatic effect on the number of predictors available for modeling?

Answer: There seems to be strong correaltion between certain predicotrs in the data as a lot of correaltion values appear between '+0.75' to '+1' and from '-0.75' to '-1' which depicts strong correlation. TO reduce the correlation between predicotrs there can be multiple ways such as elimination of certain correlated data, Principal component Analysis, etc. In the below code we give a cut off value of 0.75, after which the correlated predictors are removed from the data.

```
#3.3(c)
correlations <- cor(bbbDescr)

corrplot(correlations, order = "hclust")
```



```r
#Correlation reduction
highCorr <- findCorrelation(correlations, cutoff = .75)
length(highCorr)
```

```
## [1] 66
```

```r
highCorr
```

```
##  [1]  27  32  37  40  45  56  57  58  62  68  72  73  74  75  83  84  85
## [18]  87  88  90  93  94  95  96  97  98  99 100 101 102 103 108 110 111
## [35] 113 114 115 116 117 119 123 124 125 126 127 128   4   1  22  23  24
## [52]  33   6  49  48  21  65  67  69  80  78  89  91 109 120 112
```

```r
filteredBBData <- bbbDescr[, -highCorr]

NewBlood <- bbbDescr[ -c(27,32,37,40,45,56,57,58,62,68,72,73,74,75,83,84,85,87,88, 90 ,93,94,95,96,97,98,99,100, 101, 102, 103, 108, 110, 111, 113, 114, 115, 116, 117, 119, 123, 124, 125, 126, 127, 128, 4,1,22, 23, 24, 33, 6, 49, 48, 21, 65, 67, 69, 80, 78, 89, 91, 109, 120, 112)]
NewBlood
```