Assignment_1

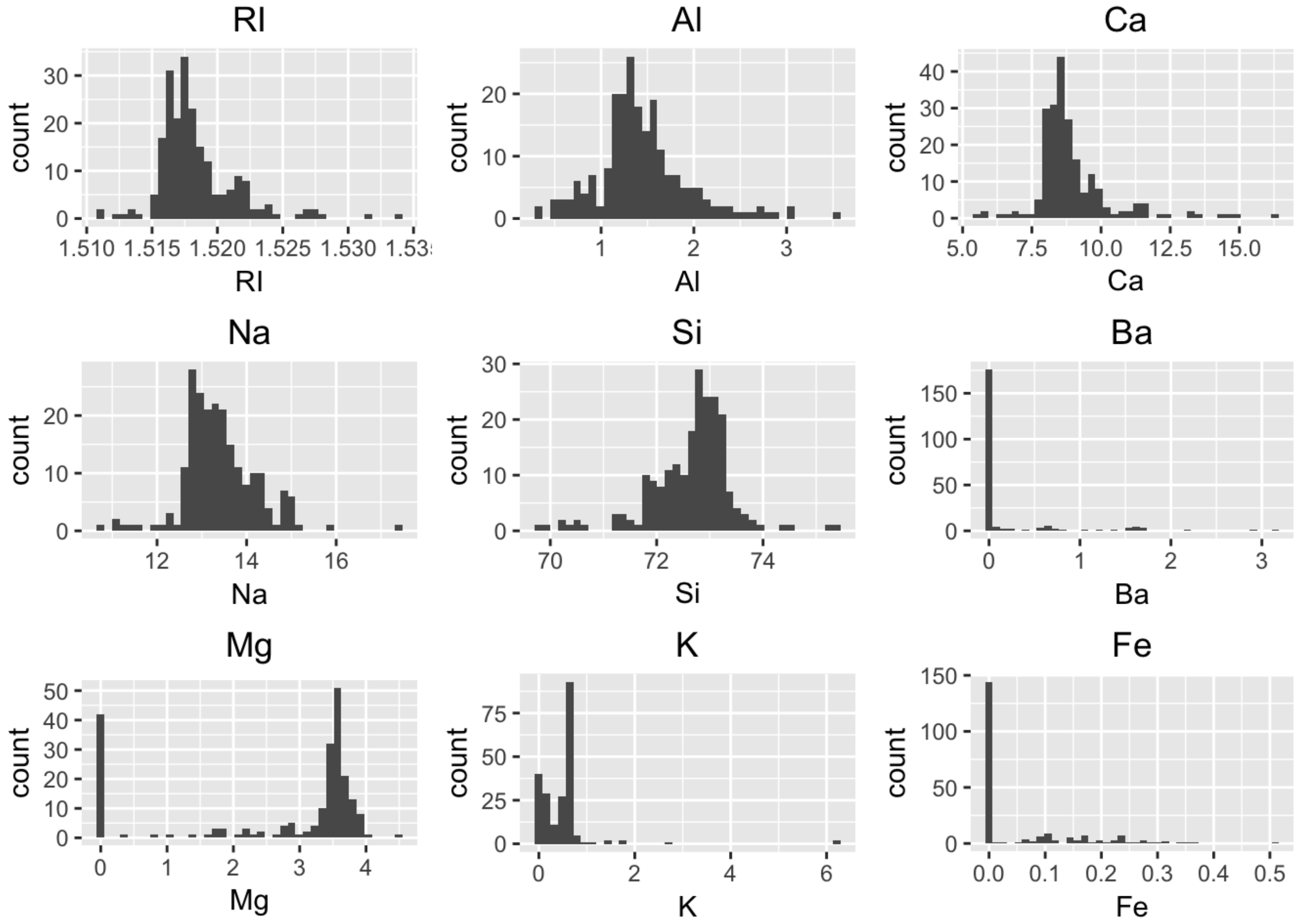
Nishanth Gandhidoss

9/22/2017

Question 1

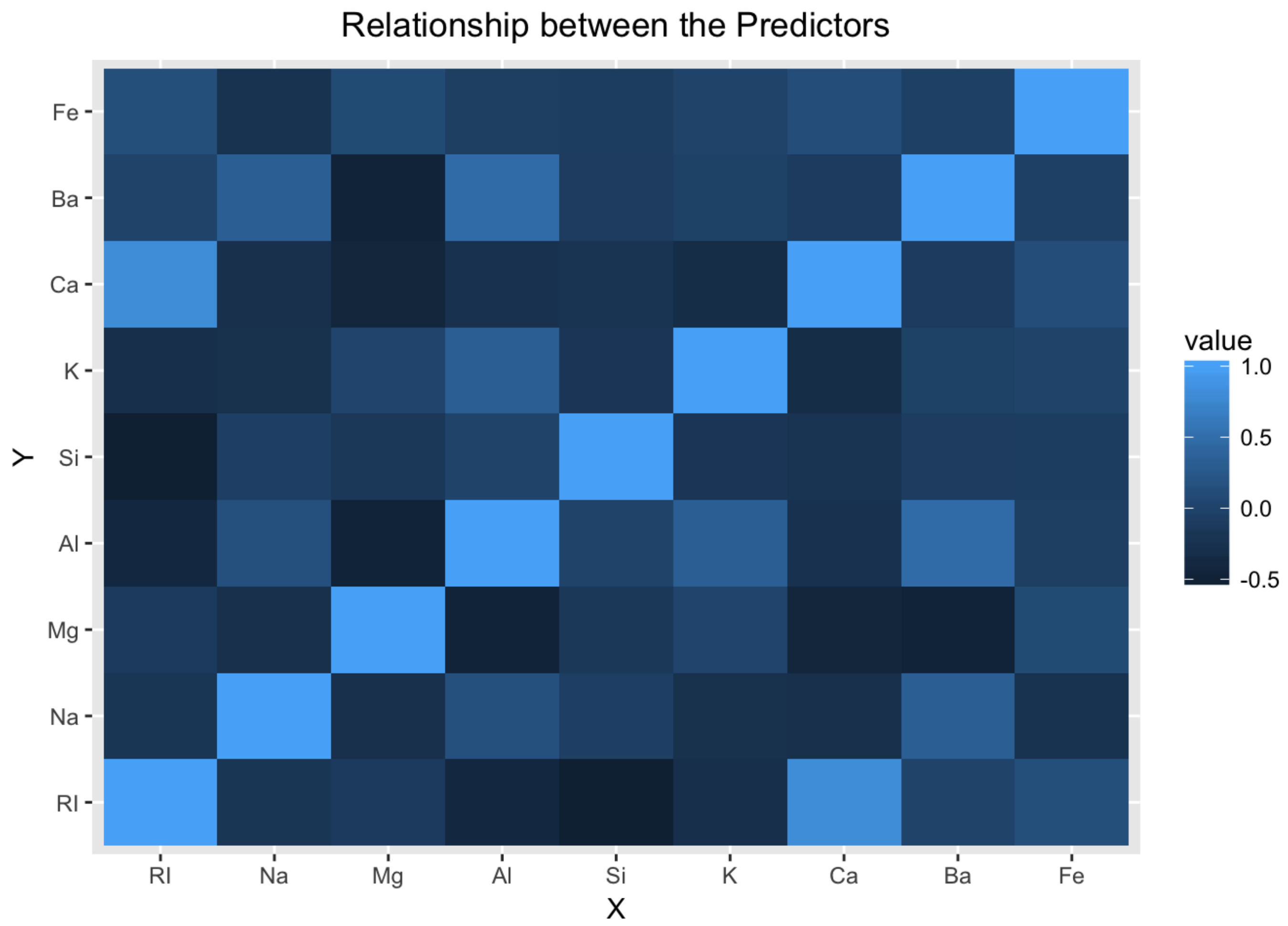
Section (a)

To examine the predictors distributions, it better to plot the predictor varaibles as seperate histograms for each. Below the histograms of the predictors distributions.



From the above figure, we can say that the predictors elements Ri, Ai, Ca, Na,Si have a distributions which are similar to the shape of the normal distribution. Although, elements like Ba, K, Fe are highly right skewed around the value zero. Mg has distribution which is more are less like a bimodal distribution with two peaks at 0 and around 3.5. Thus this is what we can learn about the distribution of the predictors.

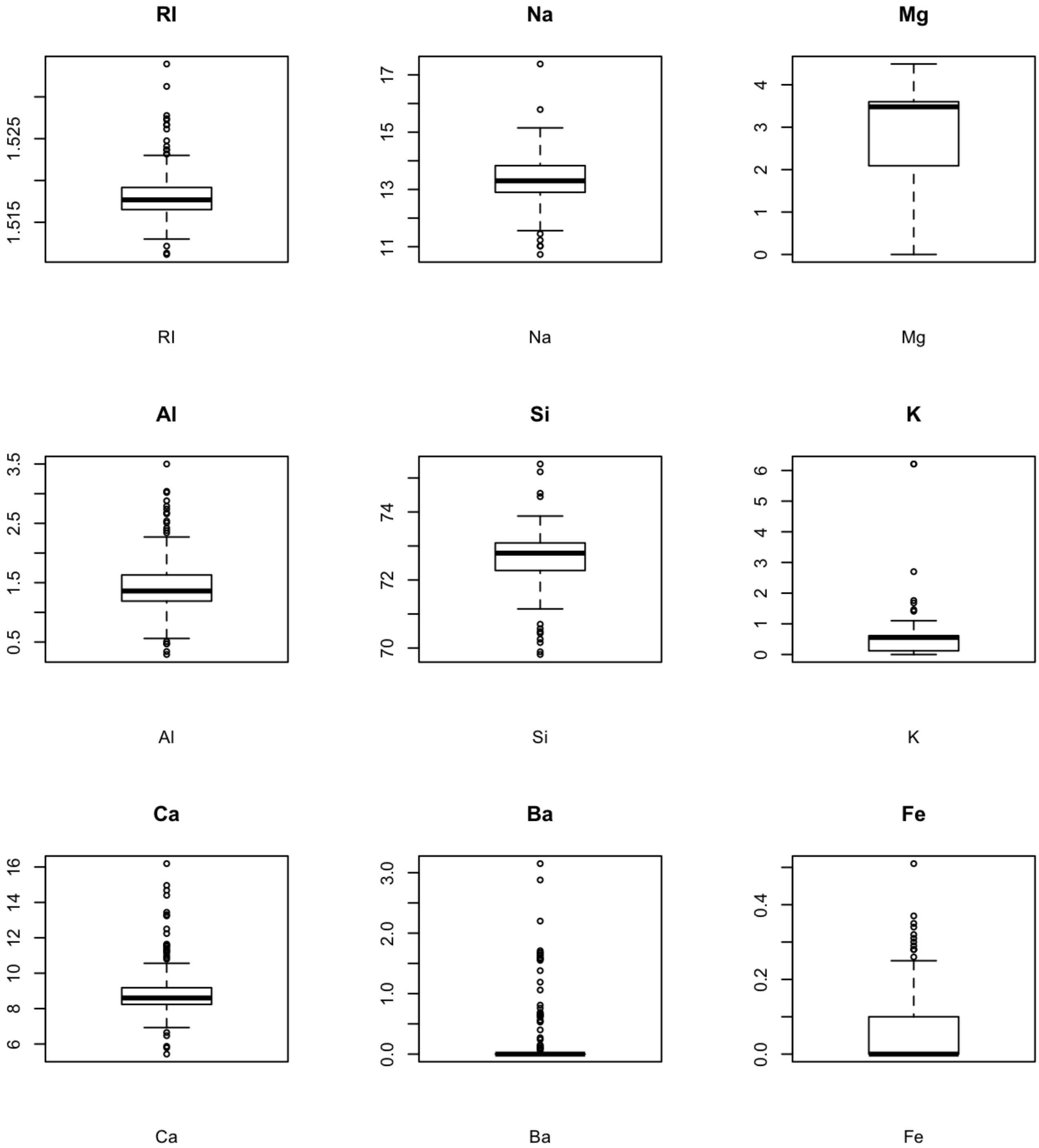
Now let us see the relationship between the predictors using the correlation plot (heatmaps).



The above correlation plot says that we have maximum neagtive probablity of -0.5. Ca and Ri looks like they have a great positive corelation. From the colors of the plot, it is evident that there are lot of negative correlation between the predictors.

Section (b)

Outliers are one of the big problem in modeling, thus it has to be identifies while the data processing stage itself in a predictive modeling life cycle process. Tha following boxplot of the predictors shows the quartile, mean, median and also the outlier information about the predictors.



Outliers in the boxplot are those small circles that are displayed outside of the horizantal lines or the 1st and 4th quartile range. There appears ot be lot of outliers in few variables expecially in Calcium and Barium. It looks like other than Mg, all other predictors are having outliers in the data. Thus it appears that there are outliers in the data. We can the skewness using the histogram itself although, we can compute the skewness value for each predictor and say how skewed they are. Those information are displayed below.

Predictor variable	Skewness Value	Skewness
RI	1.6027151	Heavily Skewed
Na	0.4478343	Symmetric
Mg	-1.1364523	Heavily Skewed
Al	0.8946104	Moderately Skewed
Si	-0.7202392	Moderately Skewed
K	6.4600889	Heavily Skewed
Ca	2.0184463	Heavily Skewed
Ba	3.3686800	Heavily Skewed
Fe	1.7298107	Heavily Skewed

Section (c)

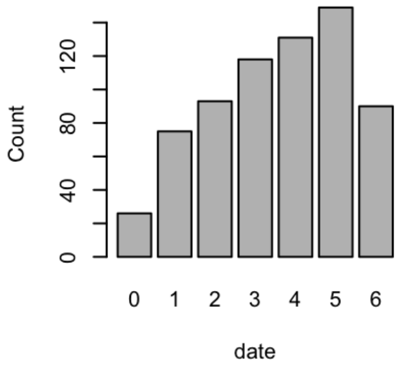
Since the predictors are mostly heavily skewed, Box Cox Transformation is a best fit to use for this predictors in order to make the model more effective. If we see the above table expect “Na” all other predictors are skewed, so applying the box cox transformation will be helpful. After applying boxcox transformation, below the results of skewness of our predictors.

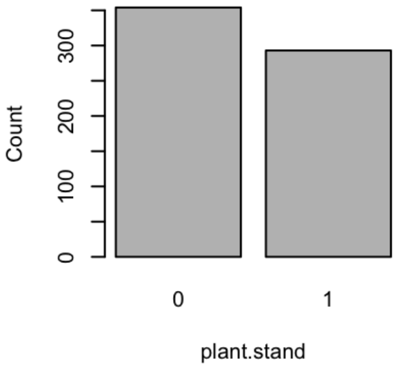
Predictor variable	Skewness Value	Skewness
RI	1.56566039	Heavily Skewed
Na	0.0338464	Symmetric
Mg	-1.13645228	Heavily Skewed
Al	0.09105899	Moderately Skewed
Si	-0.65090568	Moderately Skewed
K	6.46008890	Heavily Skewed
Ca	-0.19395573	Moderately Skewed
Ba	3.36867997	Heavily Skewed
Fe	1.72981071	Heavily Skewed

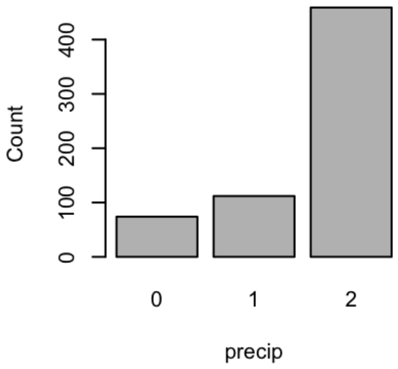
Question 2

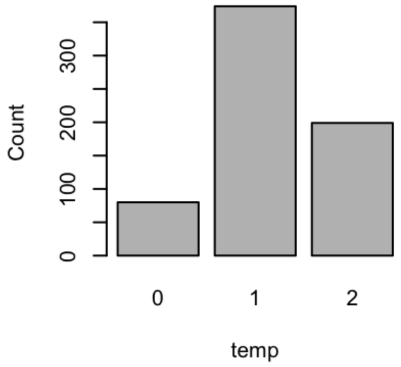
Section (a)

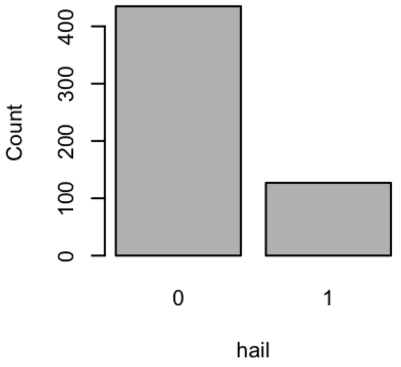
A categorical variable measures something and identifies a group to which the thing belongs. They describe a quality or characteristic of a data unit like what type or which category.Talking about the distribution of the categorical variable, we can use barplot to visualize them.

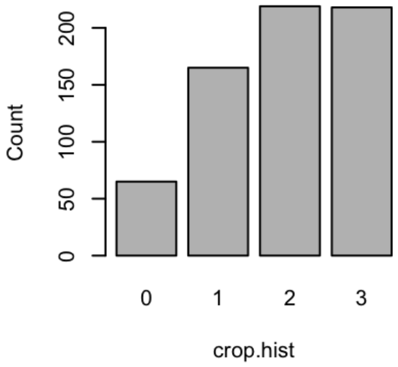

Bar chart for date

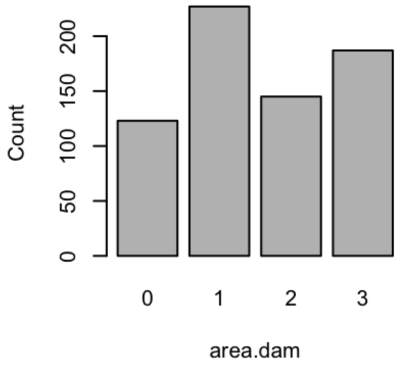

Bar chart for plant.stand

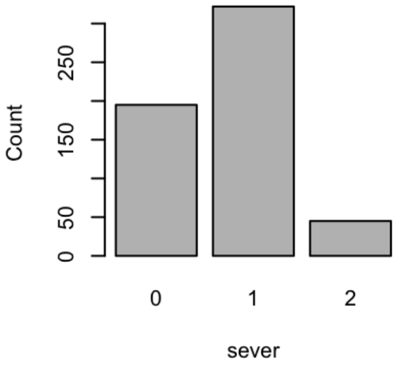

Bar chart for precip

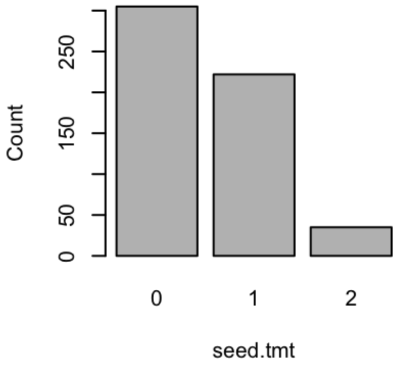

Bar chart for temp

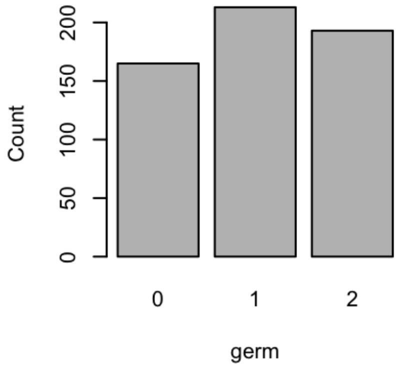

Bar chart for hail

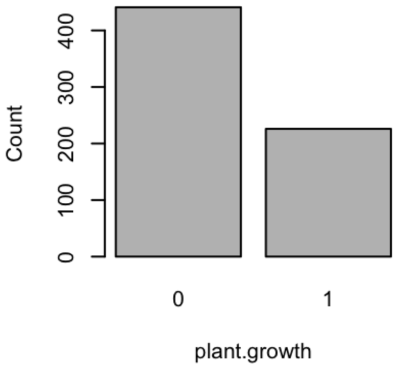

Bar chart for crop.hist

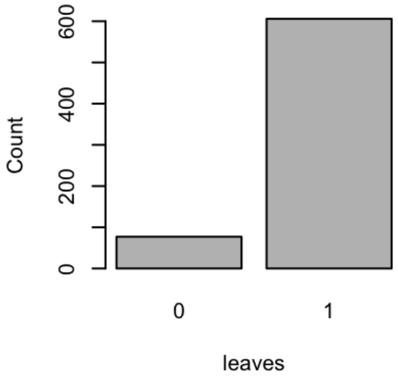

Bar chart for area.dam

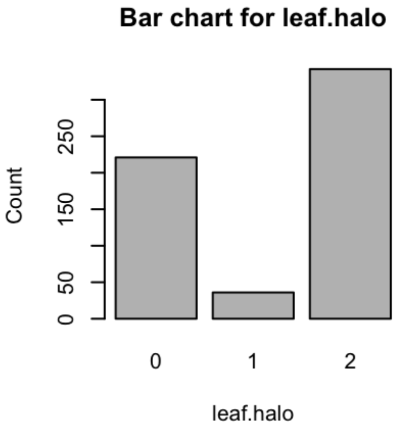

Bar chart for sever

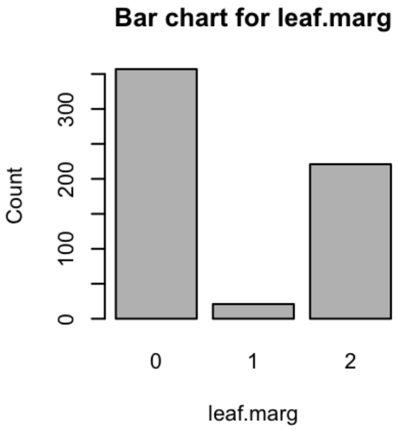

Bar chart for seed.tmt

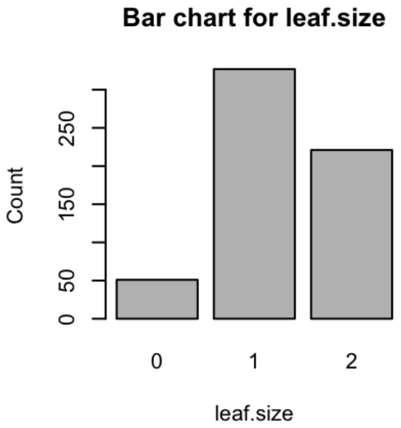

Bar chart for germ

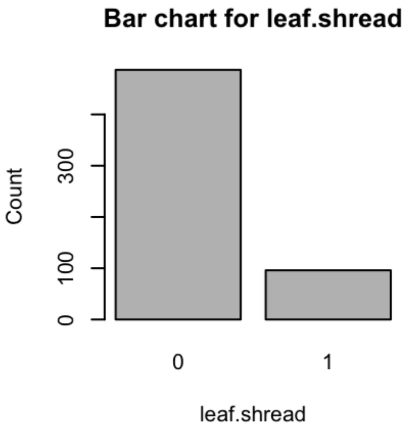

Bar chart for plant.growth

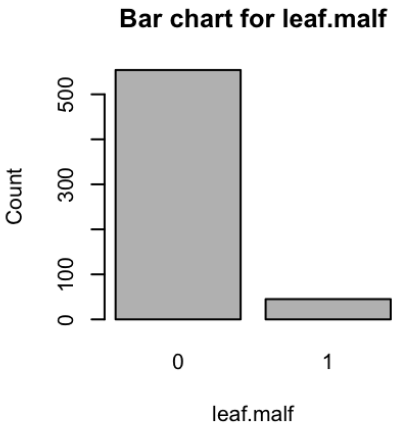

Bar chart for leaves

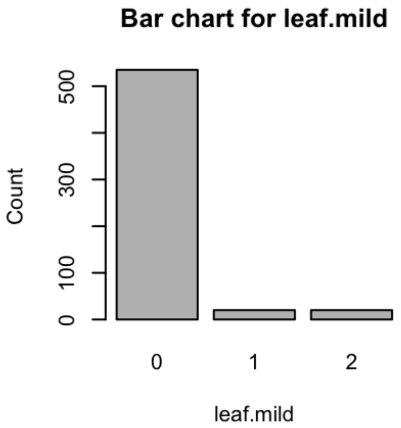

Bar chart for leaf.halo
Count
0
50
150
250
0
1
2
leaf.halo

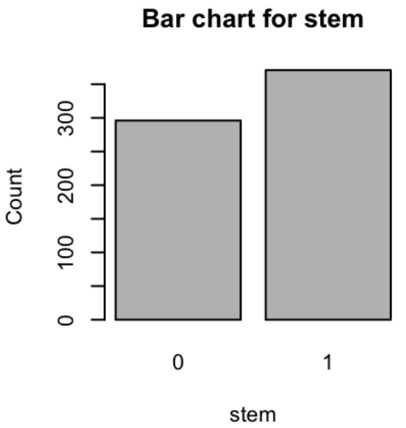

Bar chart for leaf.marg
Count
0
100
200
300
0
1
2
leaf.marg

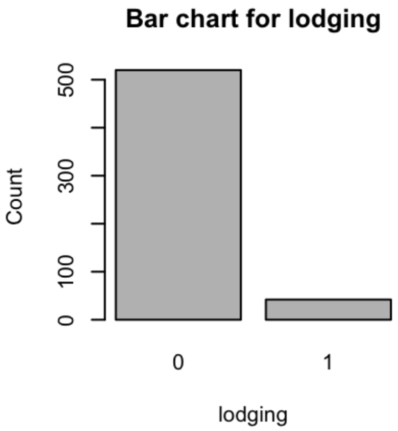

Bar chart for leaf.size
Count
0
50
150
250
0
1
2
leaf.size

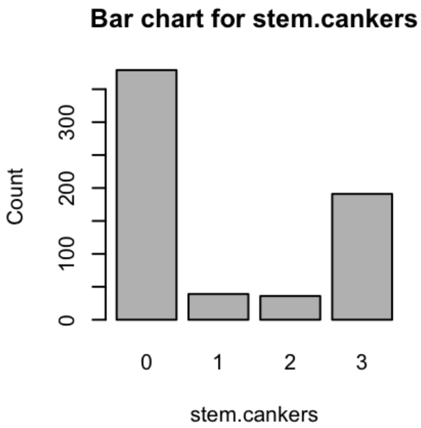

Bar chart for leaf.shread
Count
0
100
300
0
1
leaf.shread

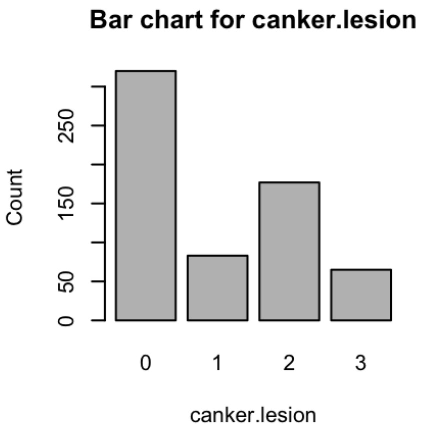

Bar chart for leaf.malf
Count
0
100
300
500
0
1
leaf.malf

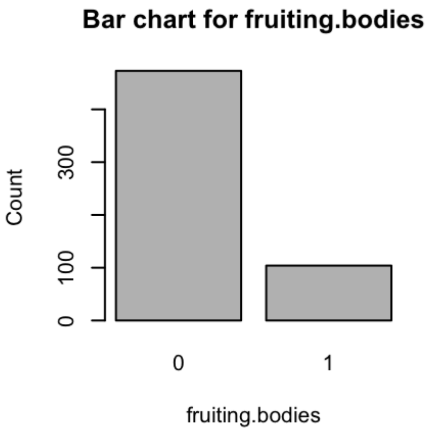

Bar chart for leaf.mild
Count
0
100
300
500
0
1
2
leaf.mild

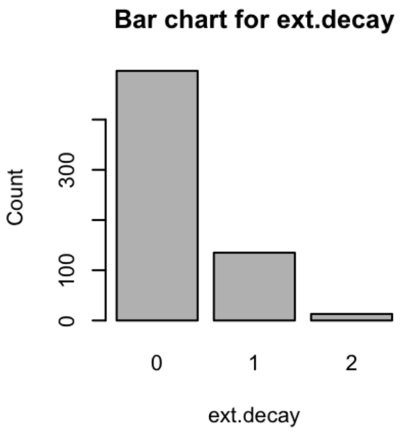

Bar chart for stem
Count
0
100
200
300
0
1
stem

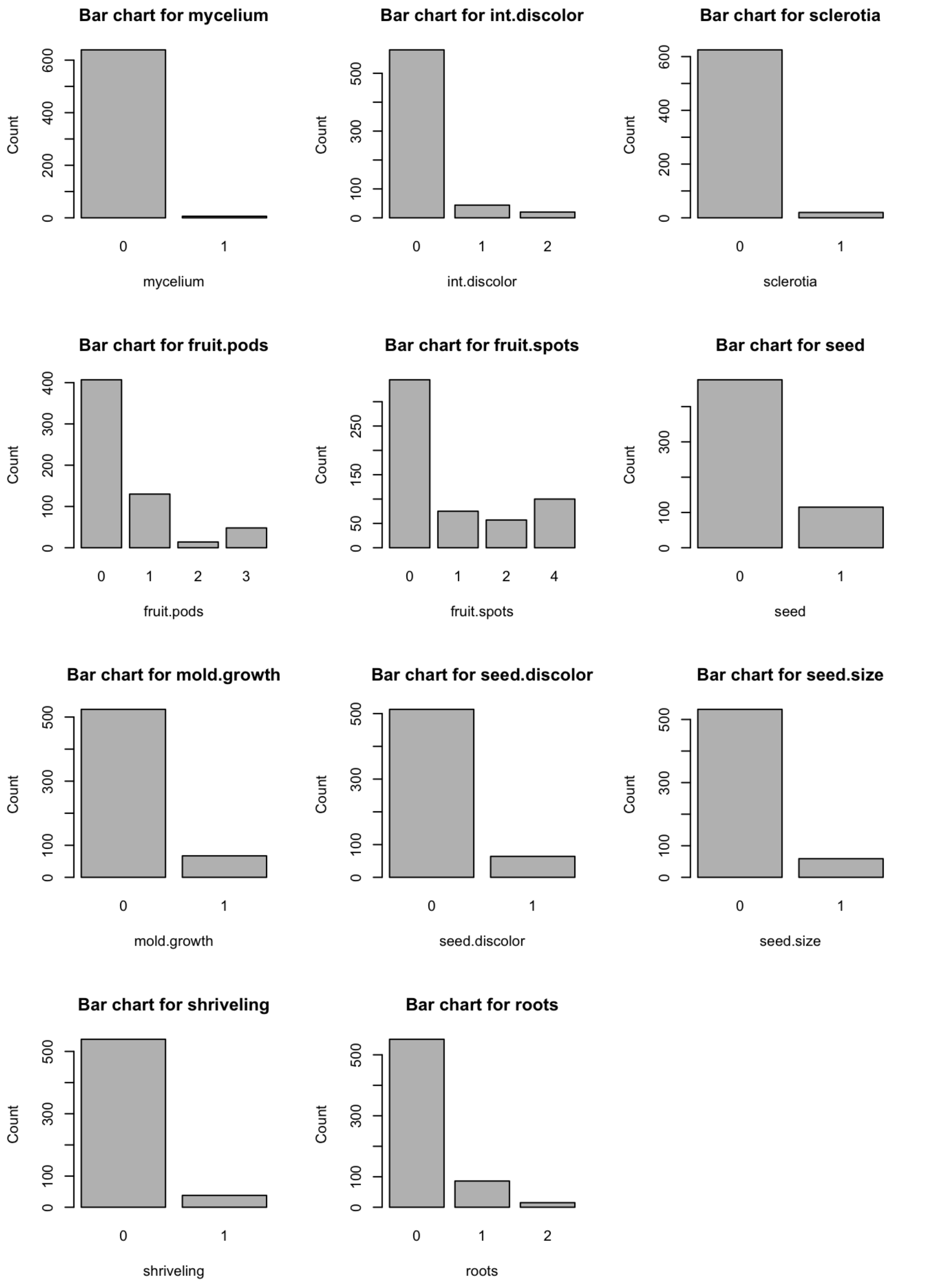

Bar chart for lodging
Count
0
100
300
500
0
1
lodging


Bar chart for stem.cankers
Count
0
100
200
300
0
1
2
3
stem.cankers


Bar chart for canker.lesion
Count
0
50
150
250
0
1
2
3
canker.lesion


Bar chart for fruiting.bodies
Count
0
100
300
0
1
fruiting.bodies


Bar chart for ext.decay
Count
0
100
300
0
1
2
ext.decay

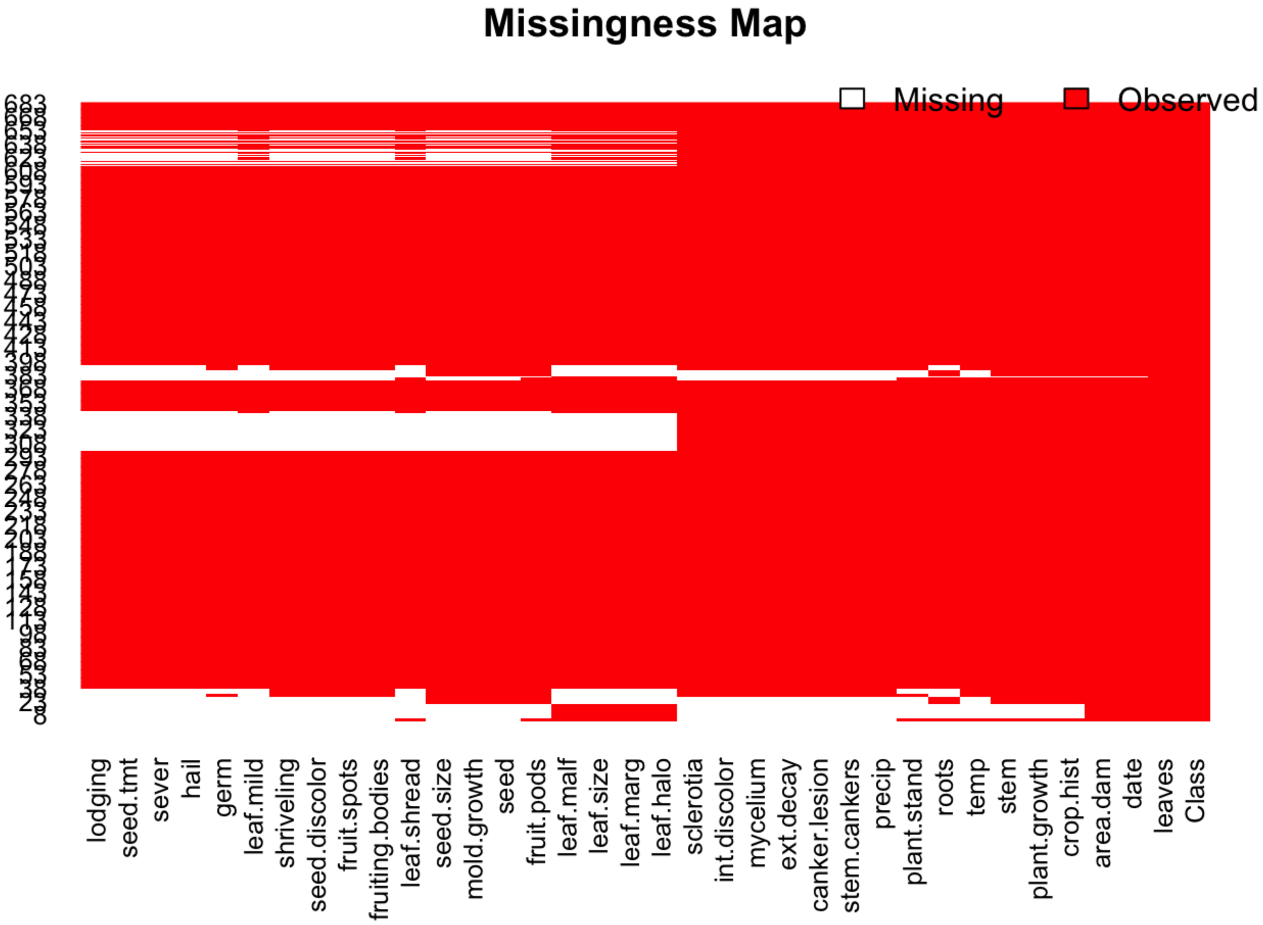


The above bar plots shows the distribution of each categorical predictors in the dataset. The best way to see is any of the predictors are degenerating is to see check the near zero variace. If there variance is almost zero then there is not going to be much effect of the categorical predictors in the model. Below the predictors with near zero variance predictiors with distributions that degenrate.

- leaf.malf
- ext.decay
- int.discolor

Section (b)

Reagrding the missing values in the predictors, the below image shows the missing values with the white places in the plot.



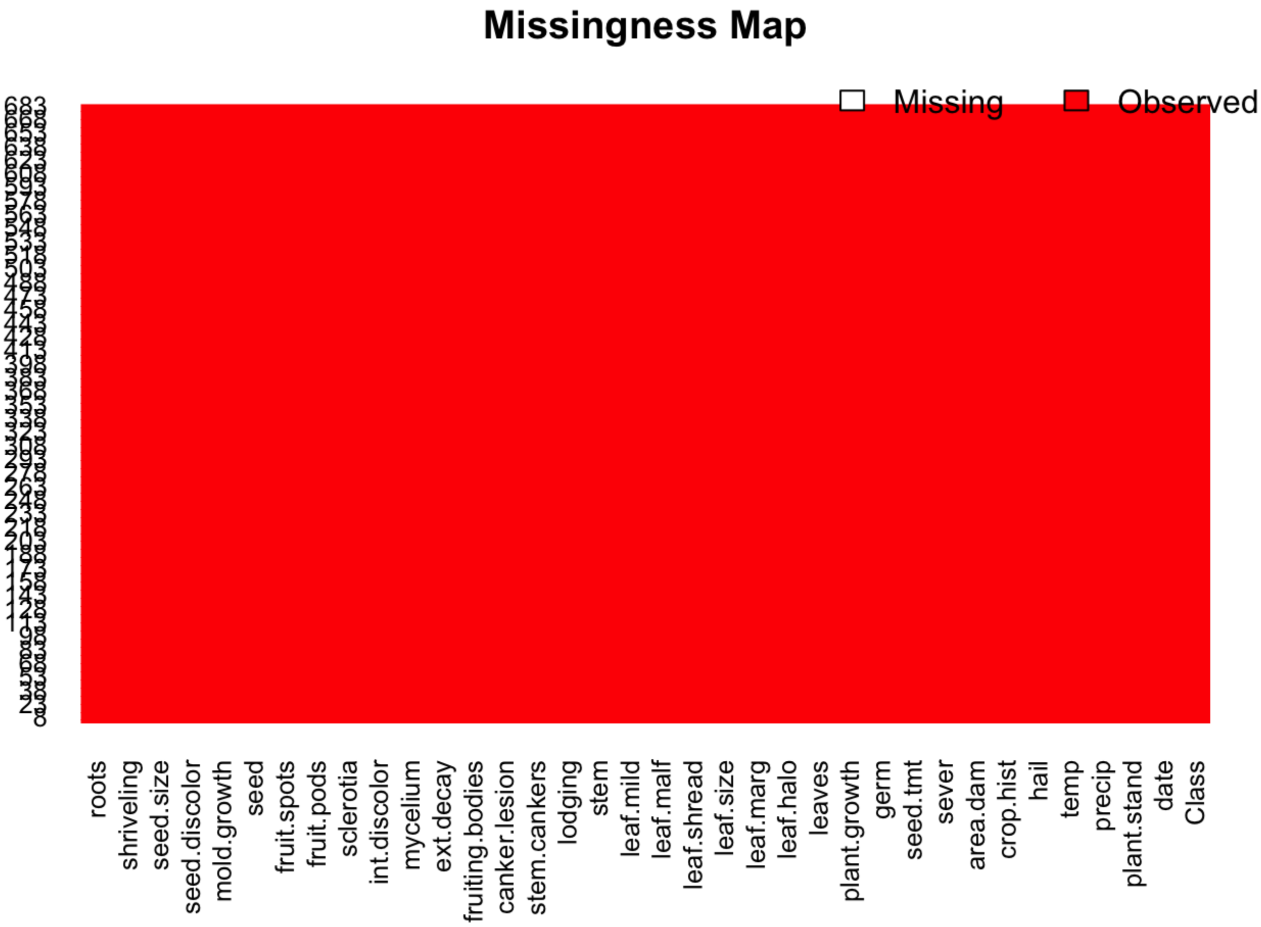
The missmap shows missing values in the dataset in white and observed values in red. The columm from lodging to leaf.halo are most likely to missing in the dataset.

```
##                               has_nans_in_sample
## Class                         FALSE TRUE
##   2-4-d-injury                    0   16
##   alternarialeaf-spot            91    0
##   anthracnose                    44    0
##   bacterial-blight               20    0
##   bacterial-pustule              20    0
##   brown-spot                     92    0
##   brown-stem-rot                 44    0
##   charcoal-rot                   20    0
##   cyst-nematode                   0   14
##   diaporthe-pod-&-stem-blight     0   15
##   diaporthe-stem-canker          20    0
##   downy-mildew                   20    0
##   frog-eye-leaf-spot             91    0
##   herbicide-injury                0    8
##   phyllosticta-leaf-spot         20    0
##   phytophthora-rot               20   68
##   powdery-mildew                 20    0
##   purple-seed-stain              20    0
##   rhizoctonia-root-rot           20    0
```

With further analysis we can see form the above results that there are many predictors completely missing for the 2-4-d-injury, cyst-nematode and herbicide-injury classes. Large amount of missing data is associated with phytophthora-rot class and moderate missing data prevails in diaporthe-pod-&-stem-blight class.

Section (c)

For NA or missing values in the predictors, lets impute the values for it using mice() in mice package. The method we are adopting for it is pmm and we will run it for 50 iterations and get 1 imputed data. Below we have missing map image of the dataset after imputation. we can see that all Na values are imputed and there is no missing values.



Question 3

Section (a)

Let us load the data from the caret package and then view some small piece of information from bbbDescr and logBBB.

```
##    tpsa nbasic
## 1 12.03      1
## 2 49.33      0
## 3 50.53      1
## 4 37.39      0
## 5 37.39      1
## 6 37.39      1
```

```
## [1]  1.08 -0.40  0.22  0.14  0.69
```

Thus the data is loaded succesfully.

Section (b)

The near zero variance function will return us the column that has variance almost equal to zero which are degenerate variables. Thus we have degenerate distributions columns in the predictors that are listed below.

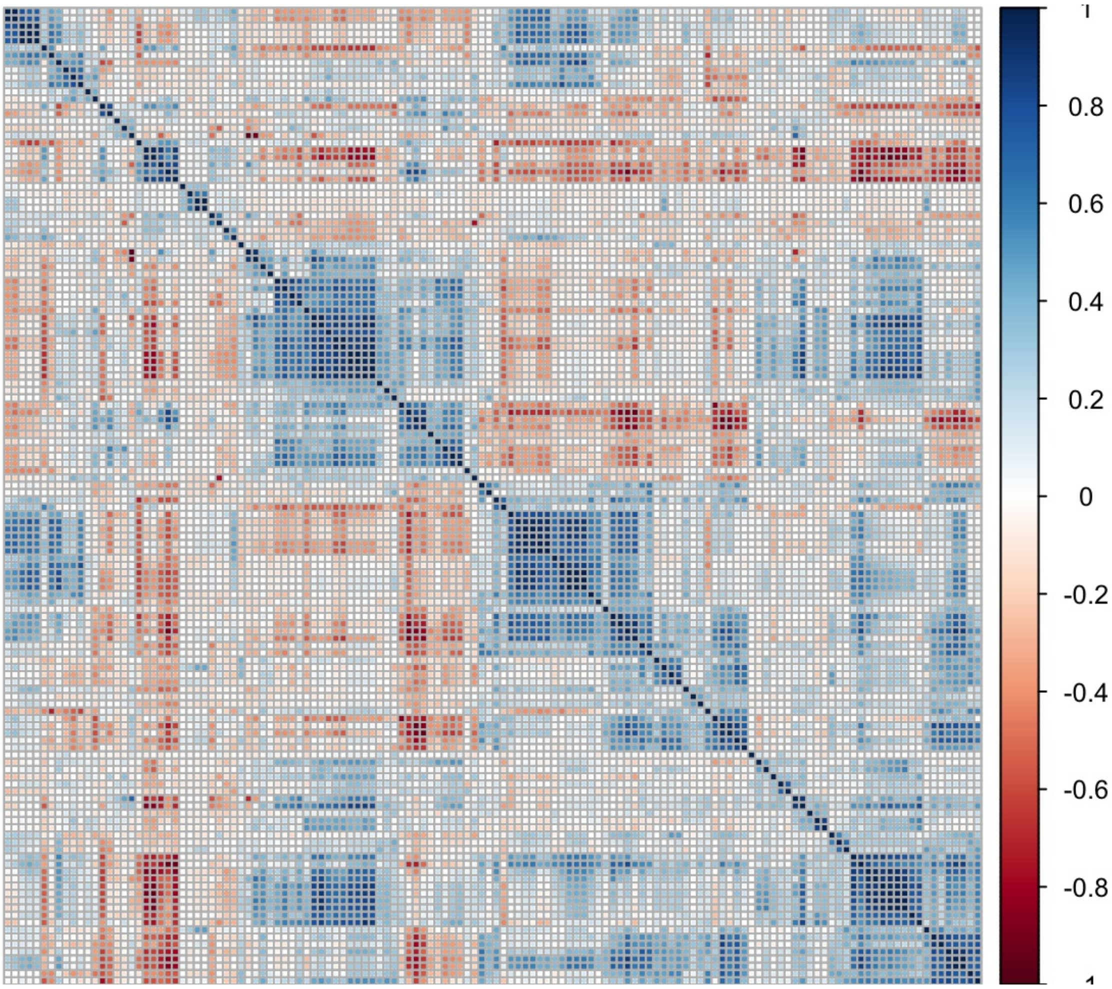
- negative

- peoe_vsa.2.1
- peoe_vsa.3.1
- a_acid
- vsa_acid
- frac.anion7.
- alert

Section (c)

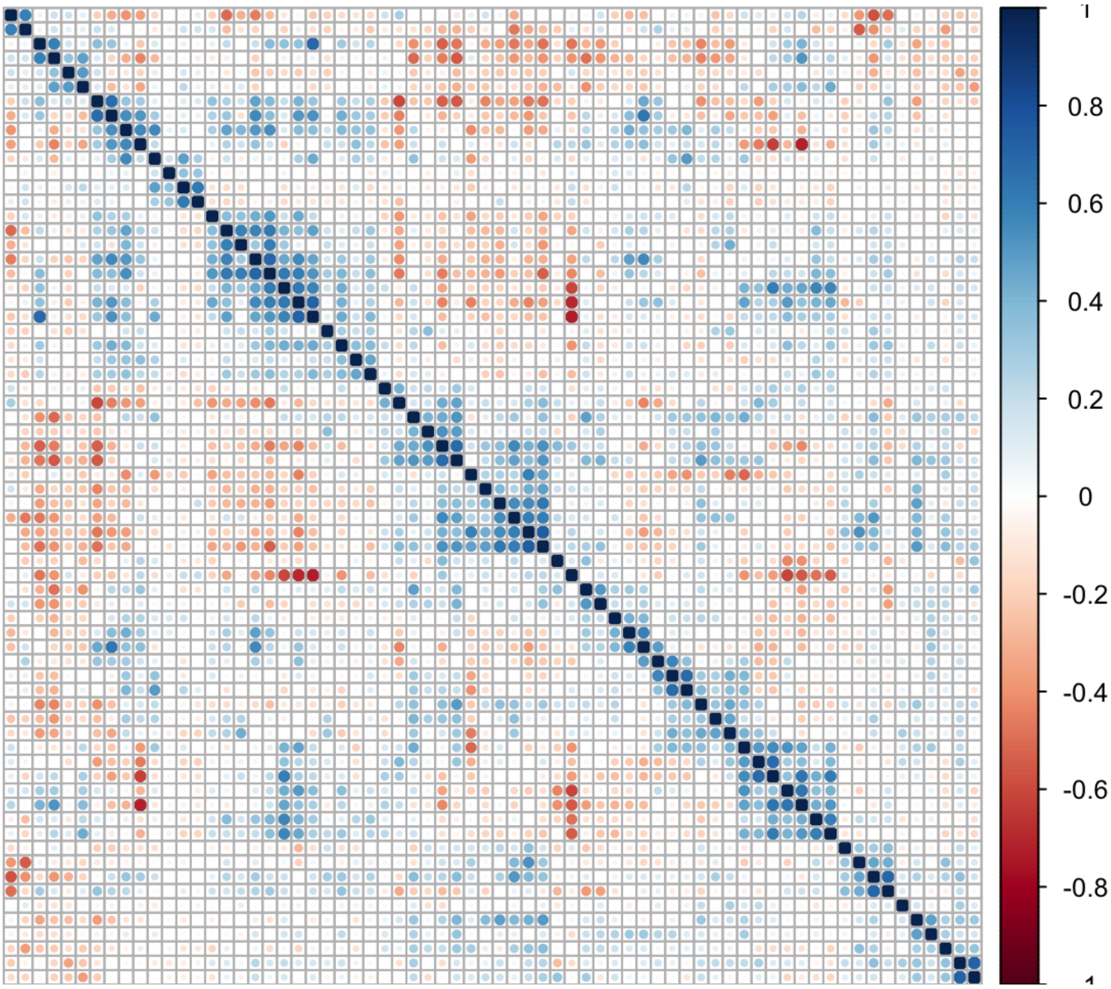
In order to find the relationship between the predictors, let us find the correlation between the predictors. We will visualize the correlation using the correlation plot or heatmap.

Relationship between Predictors



From the figure we can say that there is a lot of strong relationships between the predictors that are indcicated by the dark blue and red colours patterns. The above figure although shows the pattern of correlation between the predictors, we cannot further get much insights out of it as the number of predictors in the dataset are high in number. So in order to reduce the number of predictors in the model, one approach to use can be setting up a cut_off value (0.75) for the correlation value and only take the predictors which have absolute correlation value less than the cut off value.

Relationship between Predictors | After removing high correlated predictors



If we compare the above two plots we can see that, we have reduced the number of predictors for the model just by setting up a threshold value. Let see how many predictors were there at first and now.

- Original data contains 134 predictors
- Data after removing predictors using correlation has 68 predictors

Thus we have reduced the number of predictors avaliable for the model by removing almost half of the predictors in the dataset according to the correlation values.

*** End of Solution ***

Appendix - Coding

# Installing the packages

installNewPackage <- function(packageName) {

```
     if(packageName  %in% rownames(installed.packages()) == FALSE)

     {

               install.packages(packageName, repos = "http://cran.us.r-project.org", d
ependencies=TRUE)

     }
```

}

installNewPackage(“ggplot2”)

```
installNewPackage(“lattice”)

installNewPackage(“mlbench”)

installNewPackage(“caret”)

installNewPackage(“grid”)

installNewPackage(“gridExtra”)

installNewPackage(“reshape2”)

installNewPackage(“e1071”)

installNewPackage(“plyr”)

installNewPackage(“corrplot”)

installNewPackage(“Amelia”)

installNewPackage(“reshape2”)

installNewPackage(“mice”)

library(ggplot2)

library(lattice)

library(mlbench)

library(caret)

library(grid)

library(gridExtra)

library(reshape2)

library(e1071)

library(plyr)

library(corrplot)

library(Amelia)

library(reshape2)

library(mice)
```

Question 1

```
# Getting the data

data(Glass)

# Get the Predictors alone

pred <- Glass[,-10]

# Custom histogram function

gg_histogram <- function(col_name) {
```

```
ggplot(data = pred, aes(pred[col_name])) + ggtitle(col_name) + xlab(col_name) +

    theme(plot.title = element_text(hjust = 0.5)) + geom_histogram(bins = 40)
```

}

# Calling the user defined function

RI_hist <- gg_histogram(“RI”)

Na_hist <- gg_histogram(“Na”)

Mg_hist <- gg_histogram(“Mg”)

Al_hist <- gg_histogram(“Al”)

Si_hist <- gg_histogram(“Si”)

K_hist <- gg_histogram(“K”)

Ca_hist <- gg_histogram(“Ca”)

Ba_hist <- gg_histogram(“Ba”)

Fe_hist <- gg_histogram(“Fe”)

# Mulitplot custome function

Reference http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

multiplot <- function(…, plotlist=NULL, file, cols=1, layout=NULL) {

library(grid)

# Make a list from the … arguments and plotlist

plots <- c(list(…), plotlist)

numPlots = length(plots)

# If layout is NULL, then use ‘cols’ to determine layout

if (is.null(layout)) {

```
\# Make the panel

\# ncol: Number of columns of plots

\# nrow: Number of rows needed, calculated from \# of cols

layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),

                ncol = cols, nrow = ceiling(numPlots/cols))
```

}

if (numPlots==1) {

```
print(plots[[1]])
```

} else {

```
\# Set up the page

grid.newpage()

pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
```

# Make each plot, in the correct location

```
for (i in 1:numPlots) {

  \# Get the i,j matrix positions of the regions that contain this subplot

  matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

  print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                  layout.pos.col = matchidx$col))

}
```

}

}

# Calling the multiplot function

multiplot(RI_hist, Na_hist, Mg_hist, Al_hist, Si_hist, K_hist, Ca_hist, Ba_hist, Fe_hist, cols = 3)

# Relationship between the Predictors

cormat <- round(cor(pred), 4)

melted_cormat <- melt(cormat, varnames = c("X", "Y"))

ggplot(data = melted_cormat, aes(x=X, y=Y, fill=value)) + geom_tile() + ggtitle("Relationship between the Predictors") +

```
    theme(plot.title = element_text(hjust = 0.5))
```

# Outlier detection

par(mfrow = c(3, 3))

for(col_name in names(pred)[1:length(pred)]) {

```
boxplot(pred[col_name], main = col_name, xlab = col_name)
```

}

# Skewness

apply(pred, 2, skewness)

# Box-Cox tranformation for the predictor skewness

boxcox_skewness = function(data) {

box_cox_trans = BoxCoxTrans(data)

data_BC = predict(box_cox_trans, data)

```
skewness(data_BC)

}

# Remove the type or class label out of the dataset

Glass_predictors <- Glass[, -10]

apply(Glass_predictors, 2, boxcox_skewness)
```

Question 2

```
# Getting the data

data(Soybean)

# Set the screen layout for plotting

par(mfrow = c(3, 3))

# Loop through the categorical variables of the data

for(col in names(Soybean)[2:10]) {
```

```
barplot(table(Soybean[col]), main = paste("Bar chart for", col, "variable"), xlab =
col, ylab = "Count", col = "grey")
```

```
}

for(col in names(Soybean)[11:19]) {
```

```
barplot(table(Soybean[col]), main = paste("Bar chart for", col, "variable"), xlab =
col, ylab = "Count", col = "grey")
```

```
}

for(col in names(Soybean)[20:28]) {
```

```
barplot(table(Soybean[col]), main = paste("Bar chart for", col, "variable"), xlab =
col, ylab = "Count", col = "grey")
```

```
}

for(col in names(Soybean)[29:dim(Soybean)[2]]) {
```

```
barplot(table(Soybean[col]), main = paste("Bar chart for", col, "variable"), xlab =
col, ylab = "Count", col = "grey")
```

```
}

# Get categorical column numbers

# For all categorical predictors, need to recall the data

SoybeanCat<-Soybean[, 2:dim(Soybean)[2]]

# Calculating Near zero varaince

names(Soybean)[nearZeroVar(SoybeanCat)]

# Missing map
```

```
missmap(Soybean, col = c(“white”, “red”))

# Duplicate the data

Soybean1 <- Soybean

# Check for NA across the row wise

Soybean1$has_nans_in_sample = apply(Soybean[,-1], 1, function(x){sum(is.na(x)) > 0 })

# Tabulate the result

table(Soybean1[, c(1,37)])

# Get the imputed data using mice

imputed <- mice(Soybean, m=1, maxit = 50, method = ‘pmm’, seed = 500)

imputed_data <- complete(imputed, 1)

# Missing map

missmap(imputed_data, col = c(“white”, “red”))
```

Question 3

```
# Loading the library

library(caret)

# Data is loaded

data(BloodBrain)

bbbDescr[, c(1, 2)]

logBBB[1:5]

# Look for degenerate columns

zero_cols = nearZeroVar(bbbDescr)

colnames(bbbDescr)[zero_cols]

# Plot the correlation plot

corrplot(cor(bbbDescr), order=“hclust” )

# Finding out which predictors to elliminate since they have too large correlations

highCorr = findCorrelation(cor(bbbDescr), cutoff=0.75 )

bbbDescr_independent = bbbDescr[,-highCorr]

# Matrix has no values > cutoff=0.75

corrplot(cor(bbbDescr_independent))

# Find the number of columns in the dataset before and after the doing correlation

ncol(bbbDescr)

ncol(bbbDescr_independent)
```

End of the Assignemnt