Deanna Springgay

Ch 3 Homework

3.1

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

To begin, I graphed the predictor variables as violin plots to see the density of the data for each predictor variable.

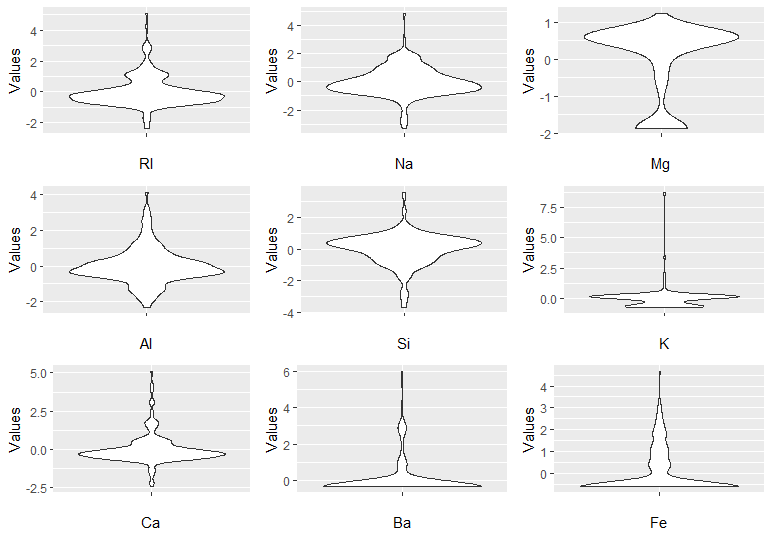


Figure 1: Violin Plots of Each Glass Predictor Variable

From Figure 1, RI, Na, Al, Si, and Ca are all approximately normal while K, Ba, and Fe are right-skewed. Mg is the oddball of left-skewed. To explore the relationships between the predictor variables, I used a correlation matrix and it is apparent that there are a few pairs of strong correlations: RI and Ca, RI and Si being the two most correlated.

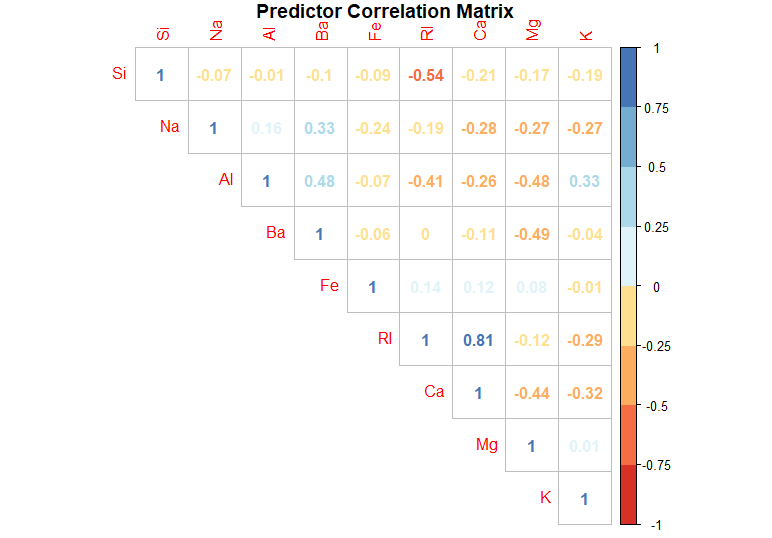


Figure 2: Correlation Matrix of Glass Predictor Variables

b) Do there appear to be any outliers in the data? Are any predictors skewed? (Please calculate the skewness values for the predictors, summarize these values using a table with interpretations).

Within the predictor values, every predictor except for Magnesium contain outliers. All the predictors except for Sodium show skewness, as indicated by the following table:

|  |  |  |
| --- | --- | --- |
| **Predictor** | **Skewness** | **Interpretation** |
| RI | 1.6027151 | Highly Right Skewed |
| Na | 0.4478343 | Approximately Symmetric |
| Mg | -1.1364523 | Highly Left Skewed |
| Al | 0.8946104 | Moderately Right Skewed |
| Si | -0.7202392 | Moderately Left Skewed |
| K | 6.4600889 | Highly Right Skewed |
| Ca | 2.0184463 | Highly Right Skewed |
| Ba | 3.3686800 | Highly Right Skewed |
| Fe | 1.7298107 | Highly Right Skewed |

c) Are there any relevant transformations of one or more predictors that might improve the classification model? (Please perform at least two transformations based on your observations of the predictors; use visualizations of before and after the transformations; and make comments).

To reduce the skewness seen in the predictors, I decided to transform the data using BoxCox to reduce the skewness in individual predictors and PCA to reduce the skewness, correlation between the predictors, and the number of predictors. Figures 3-5 show the relationships between variables and Figures 6-8 show the correlation matrices.

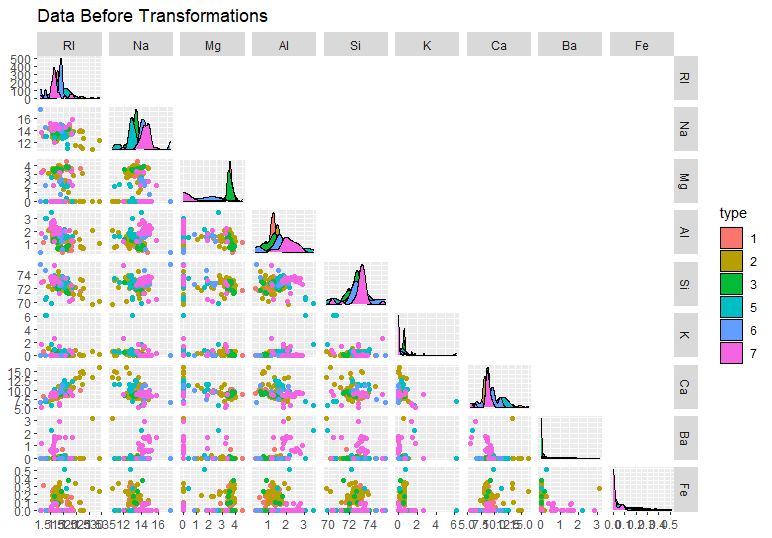


Figure 3: Scatter Plot Matrix of Pre-Processed Data

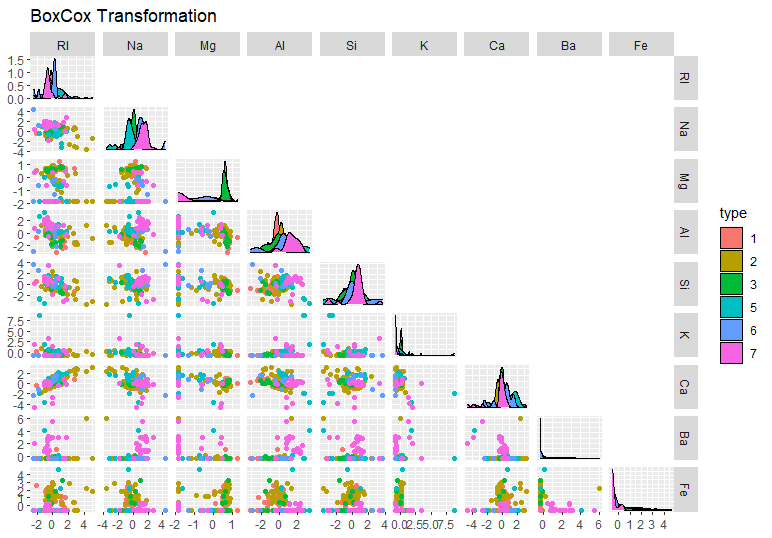


Figure 4: Scatter Plot Matrix of BoxCox Transformation

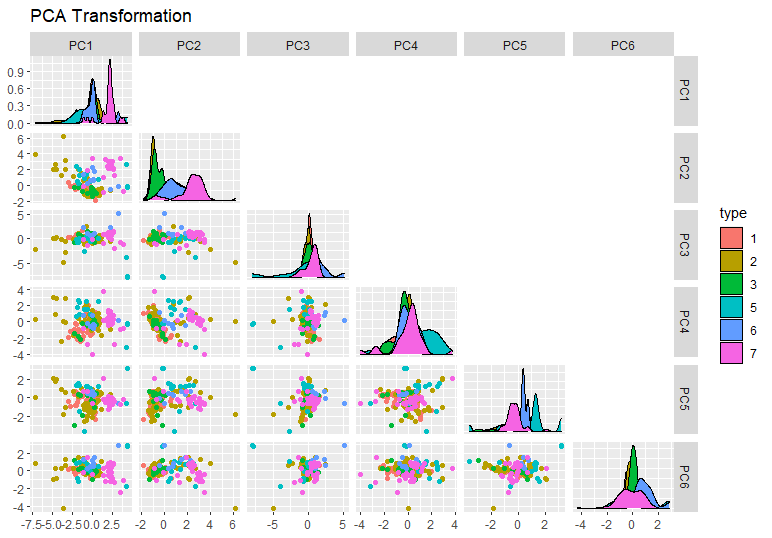


Figure 5: Scatter Plot Matrix of PCA Transformation

Before taking a look at the correlation matrices of the transformed data, Figures 3-5 show the scatterplot matrices of each to see how each predictor was affected. Comparing the preprocessed data to the BoxCox transformation, the only big difference would be the axes: the points have all shifted towards zero yet the relative locations of the points does not appear to have any significant difference. When we compare the PCA transformation to the preprocessed data, not only have the data been centered but the data points are also more clustered together in groups.

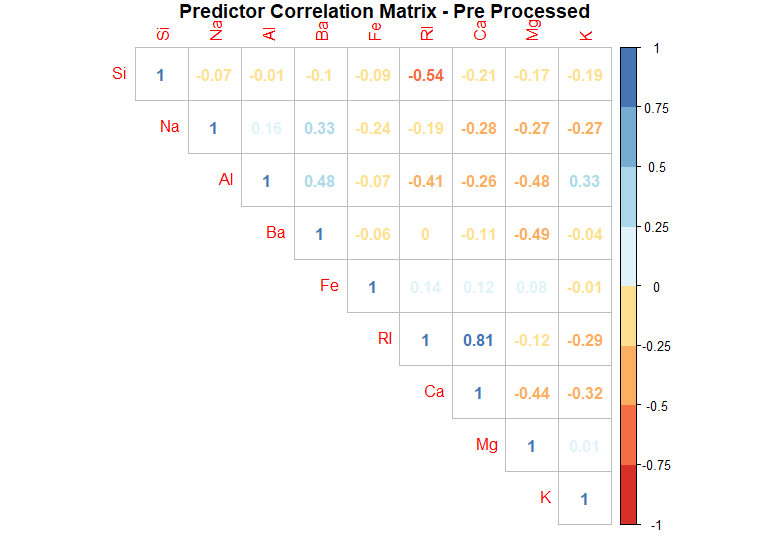


Figure 6: Correlation Matrix of Pre-Processed Data

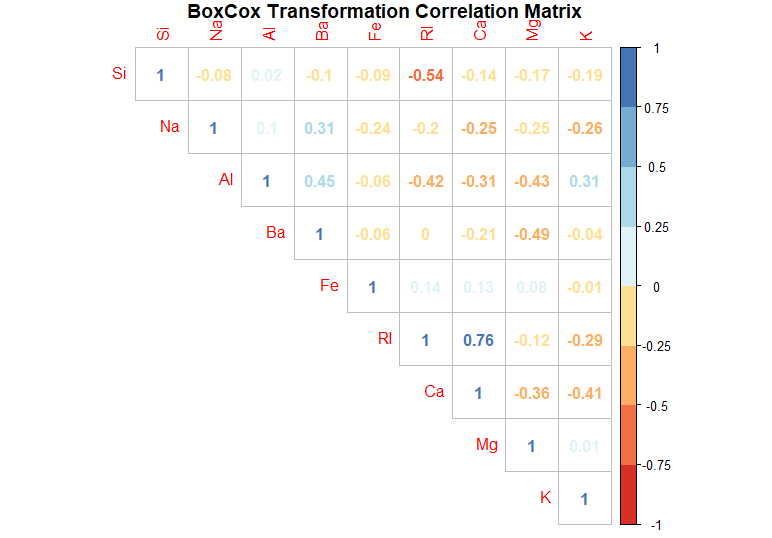


Figure 7: Correlation Matrix of BoxCox Trasformation

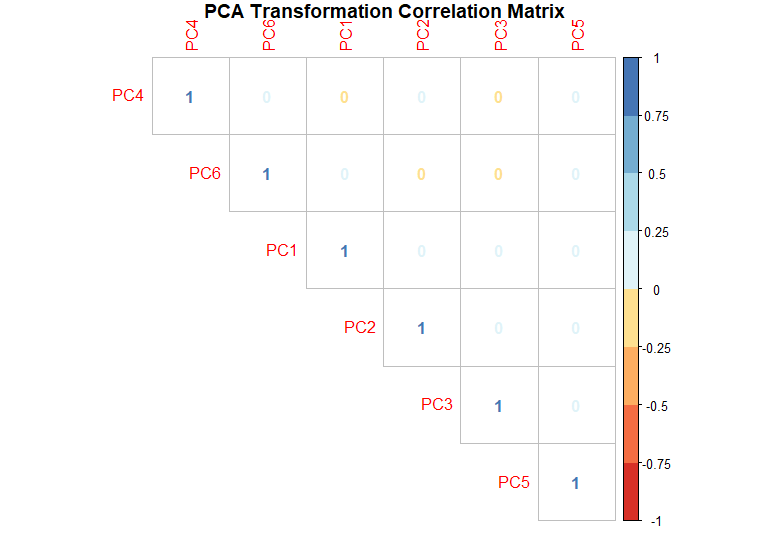


Figure 8: Correlation Matrix of PCA Transformation

Finally, looking at the correlation matrices indicates that BoxCox made slight corrections to correlations but the PCA transformation reduced all correlations to zero.

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?

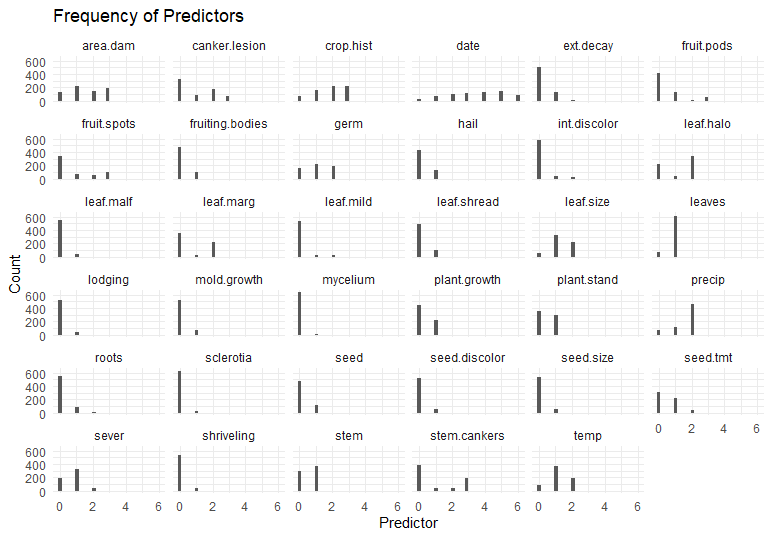


Figure : Frequency of Soybean Predictors

Visually looking over the distributions of the Soybean predictors, it seems that this dataset would be a good candidate for removing degenerate predictors. Running the nearZeroVar functions returns three degenerate predictors: leaf.mild, mycelium, and sclerotia.

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?

Taking a look at Figure 10, the missing data indicators follow a vertical pattern which depicts a sample that is missing data versus a predictor that is missing data. In addition to that, the missing data are in groups together over sections of the same or similar Class. The Classes with missing data seem to be phytophthora-rot, diaporthe-pod-&-stem-blight, cyst-nematode, 2-4-d-injury, and herbicide-injury.

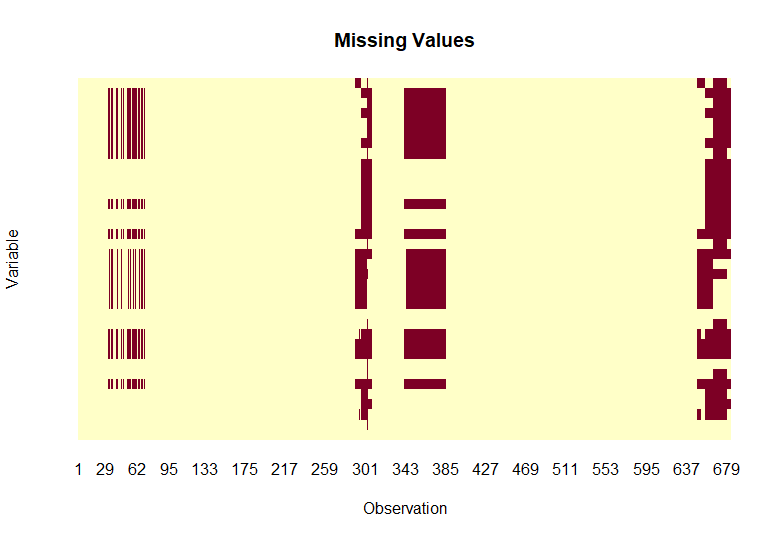


Figure : Soybean Missing Values

c) Develop a strategy for handling missing data, either by eliminating predictors or imputation. (You only need to provide the strategy, do not need to implement the strategy).

Since the missing data is not associated with particular predictors, I would approximate the missing data using imputation. This could easily be inaccurate, as significant data for the affected Classes are missing. If the associated predictors were eliminated, a far larger amount of data would be removed and any model built on the remaining data would be potentially quite useless.

3.3

a) Start R and use these commands to load the data.

b) 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?

Looking at Figure 11, there are many predictors that are correlated to a high degree. Considering the high number of predictors, a model would be easier to calculate by using less redundant predictor variables. After running the findCorrelation function and removing the associated predictors, the total number of predictors is reduced to 85. Considering there are 134 individual samples, having 85 predictors Is acceptable and reduces the computational power required to further compute a model.

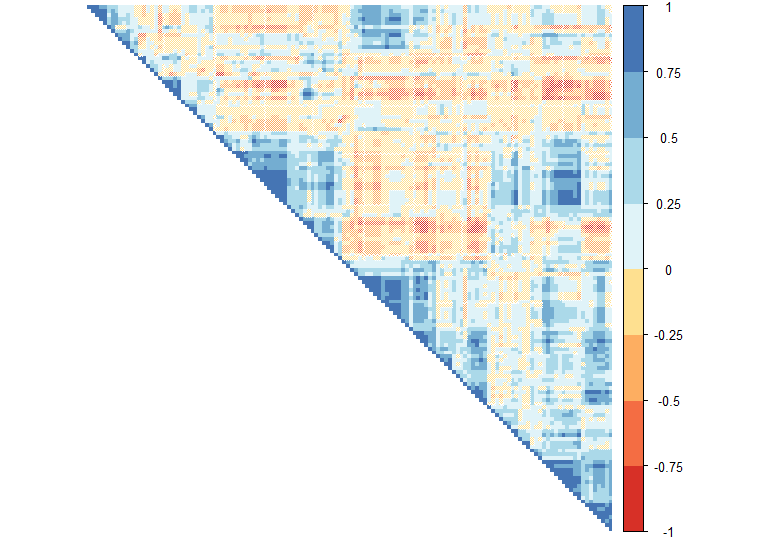


Figure : Correlation Matrix of BloodBrain Predictors

R Code

library(AppliedPredictiveModeling)

library(caret)

library(corrplot)

library(e1071)

library(GGally)

library(ggpubr)

library(mlbench)

library(RColorBrewer)

library(tidyverse)

####Exercise 3.1####

#first the data is loaded

data(Glass)

#next the 'Type' column needs to be removed from the dataset,

#there is no need to remove more columns as the predictors are all continuous

type <- Glass[,10]

clean\_glass <- Glass[,1:9]

dirty\_glass <- Glass[,1:9]

#centering and scaling the data

clean\_glass <- clean\_glass %>%

scale(center = TRUE, scale = TRUE) %>%

as.data.frame()

#to visualize the data, violinplots of each predictor will be made into one plot

ri\_plot <- ggplot(clean\_glass, aes(x="",y=RI)) +

geom\_violin() +

labs(x = "RI", y = "Values")

na\_plot <- ggplot(clean\_glass, aes(x="",y=Na)) +

geom\_violin() +

labs(x = "Na", y = "Values")

mg\_plot <- ggplot(clean\_glass, aes(x="",y=Mg)) +

geom\_violin() +

labs(x = "Mg", y = "Values")

al\_plot <- ggplot(clean\_glass, aes(x="",y=Al)) +

geom\_violin() +

labs(x = "Al", y = "Values")

si\_plot <- ggplot(clean\_glass, aes(x="",y=Si)) +

geom\_violin() +

labs(x = "Si", y = "Values")

k\_plot <- ggplot(clean\_glass, aes(x="",y=K)) +

geom\_violin() +

labs(x = "K", y = "Values")

ca\_plot <- ggplot(clean\_glass, aes(x="",y=Ca)) +

geom\_violin() +

labs(x = "Ca", y = "Values")

ba\_plot <- ggplot(clean\_glass, aes(x="",y=Ba)) +

geom\_violin() +

labs(x = "Ba", y = "Values")

fe\_plot <- ggplot(clean\_glass, aes(x="",y=Fe)) +

geom\_violin() +

labs(x = "Fe", y = "Values")

ggarrange(ri\_plot, na\_plot, mg\_plot, al\_plot, si\_plot, k\_plot, ca\_plot, ba\_plot, fe\_plot)

#checking the skewness of the predictors

apply(clean\_glass, 2, skewness)

#next is a correlation matrix to show the relationships between predictors

corrplot(cor(clean\_glass),

method = "number",

type = "upper",

order = "hclust",

col = brewer.pal(n=8, name="RdYlBu"),

title = "Predictor Correlation Matrix",

mar=c(0,0,1,0))

#as for transformations, all predictors except for Na would benefit from a transformation as they are moderately or highly skewed

#We will use boxcox and PCA to transform the data

glass\_1 <- dirty\_glass

glass\_2 <- dirty\_glass

box\_transformation <- predict(preProcess(glass\_1, method = c("center", "scale", "BoxCox")), glass\_1)

pca\_transformation <- predict(preProcess(glass\_2, method = c("center", "scale", "pca")), glass\_2)

dirty\_glass$type <- type

box\_transformation$type <- type

pca\_transformation$type <- type

#and now for visualizing the transformations along with correlation matrices for further comparison

ggpairs(dirty\_glass,

columns = 1:9,

ggplot2::aes(color = type),

upper = list(continuous='blank'),

legend=1,

title = "Data Before Transformations")

ggpairs(box\_transformation,

columns = 1:9,

ggplot2::aes(color = type),

upper = list(continuous='blank'),

legend=1,

title = "BoxCox Transformation")

ggpairs(pca\_transformation,

columns = 1:6,

ggplot2::aes(color = type),

upper = list(continuous='blank'),

legend=1,

title = "PCA Transformation")

corrplot(cor(dirty\_glass[,1:9]),

method = "number",

type = "upper",

order = "hclust",

col = brewer.pal(n=8, name="RdYlBu"),

title = "Predictor Correlation Matrix - Pre Processed",

mar=c(0,0,1,0))

corrplot(cor(box\_transformation[,1:9]),

method = "number",

type = "upper",

order = "hclust",

col = brewer.pal(n=8, name="RdYlBu"),

title = "BoxCox Transformation Correlation Matrix",

mar=c(0,0,1,0))

corrplot(cor(pca\_transformation[,1:6]),

method = "number",

type = "upper",

order = "hclust",

col = brewer.pal(n=8, name="RdYlBu"),

title = "PCA Transformation Correlation Matrix",

mar=c(0,0,1,0))

####Exercise 3.2####

data("Soybean")

#remove ordered factors and compensated for everything increased by 1 during conversion

Soybean <- Soybean %>%

mutate\_at(vars(`date`:`roots`), as.numeric)

Soybean[,2:36] <- Soybean[,2:36] - 1

#make the data long for plotting

long\_soy <- Soybean %>%

pivot\_longer(cols = c(date:roots), names\_to = "predictor", values\_to = "values")

#plotting!

ggplot(long\_soy, aes(x = values)) +

geom\_histogram() +

facet\_wrap(~predictor) +

theme\_minimal() +

labs(title = "Frequency of Predictors", x = "Predictor", y = "Count")

#delinquent predictors don't have much variation and one value is way higher than the others

nearZeroVar(Soybean)

#seeing missing values

image(is.na(Soybean), main = "Missing Values", xlab = "Observation", ylab = "Variable", xaxt = "n", yaxt = "n", bty = "n")

axis(1, seq(0, 1, length.out = nrow(Soybean)), 1:nrow(Soybean), col = "white")

#for missing data I would use imputation over removing predictors due to the pattern of the missing data.

#Sections of rows are missing versus a column of data from a particular predictor.

####Exercise 3.3####

data(BloodBrain)

#preprocessing data for correlations

transformations <- bbbDescr %>%

preProcess(method = c("center", "scale"))

clean\_brain <- predict(transformations, bbbDescr)

#plotting a correlation matrix, which is an absolute nightmare

correlation <- cor(clean\_brain)

corrplot(correlation, method = "shade", order = "hclust", type = "upper", col = brewer.pal(n=8, name="RdYlBu"), tl.pos = 'n')

#here's what correlated predrictors should be removed

high\_cor <- findCorrelation(correlation, cutoff = .85)

remaining\_predictors <- ncol(clean\_brain) - length(high\_cor)

remaining\_predictors