HW3 For Applied Data Mining
STAT W 3026-4026
Spring 2016
Columbia University

Robin Lee
rcl2136
QMSS MA

February 11, 2016

1 Exercise 3.1

The UC Irvine Machine Learning Repository6 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.

The data can be accessed via:

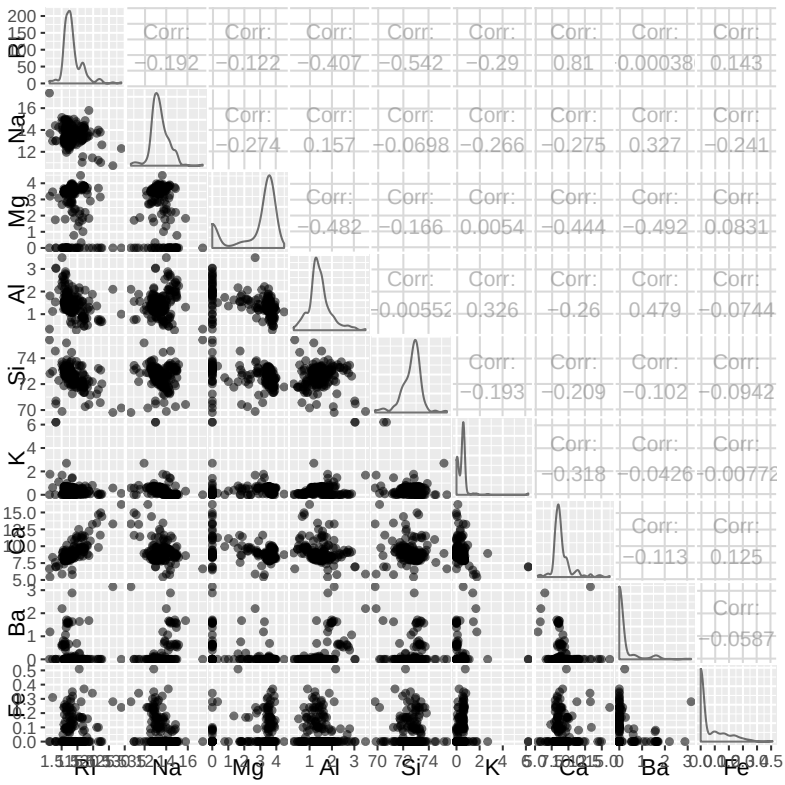
```
> library(ggplot2)
> library(GGally)
> library(dplyr)
> library(mlbench)
> data(Glass)
> glimpse(Glass)

Observations: 214
Variables: 10
$ RI   (dbl) 1.52101, 1.51761, 1.51618, 1.51766, 1.51742, 1.51596, 1.51743,...
$ Na   (dbl) 13.64, 13.89, 13.53, 13.21, 13.27, 12.79, 13.30, 13.15, 14.04,...
$ Mg   (dbl) 4.49, 3.60, 3.55, 3.69, 3.62, 3.61, 3.60, 3.61, 3.58, 3.60, 3....
```

```
$ Al   (dbl) 1.10, 1.36, 1.54, 1.29, 1.24, 1.62, 1.14, 1.05, 1.37, 1.36, 1....
$ Si   (dbl) 71.78, 72.73, 72.99, 72.61, 73.08, 72.97, 73.09, 73.24, 72.08,...
$ K    (dbl) 0.06, 0.48, 0.39, 0.57, 0.55, 0.64, 0.58, 0.57, 0.56, 0.57, 0....
$ Ca   (dbl) 8.75, 7.83, 7.78, 8.22, 8.07, 8.07, 8.17, 8.24, 8.30, 8.40, 8....
$ Ba   (dbl) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
$ Fe   (dbl) 0.00, 0.00, 0.00, 0.00, 0.00, 0.26, 0.00, 0.00, 0.00, 0.11, 0....
$ Type (fctr) 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
```

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

  My options are scatterplot matrix and histogram. Thus, I will use ggpairs in GGally package to get the plot. I also adjust the transparency to make scatter plots easier to read.

```
> ggpairs(Glass, columns= 1:9,
+         mapping = ggplot2::aes(alpha = 0.2))
```



The only two predictors that have a correlation coefficient above 0.5 or below -0.5 are RI and Ca.

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

  Yes, from the histogras, Al, K, CA, Ba, and Fe are skewed to the right. Mg is skewed to the left and has a smaller peak on the left. There are some samples with high values in K, Ca and Ba.

  I will calculate skewness to confirm.

```
> library(e1071)
> skewValues <- apply(Glass[,1:9], 2, skewness)
> skewValues

       RI         Na         Mg         Al         Si          K         Ca
1.6027151  0.4478343 -1.1364523  0.8946104 -0.7202392  6.4600889  2.0184463
       Ba         Fe
3.3686800  1.7298107
```

  The result is similar to what I got judging from the histograms.

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

  Yes, for the skewed variables, I should transform the predictors. I will use Box-cox test to find out what transformation is needed.

```
> library(caret)
> boxcoxValues = preProcess(Glass[,-10],method = "BoxCox")
> boxcoxValues

Created from 214 samples and 5 variables

Pre-processing:
  - Box-Cox transformation (5)
  - ignored (0)

Lambda estimates for Box-Cox transformation:
-2, -0.1, 0.5, 2, -1.1
```

```
> BC = apply(Glass[,-10],2, BoxCoxTrans)
> # transform variables
> GlassBC = predict(boxcoxValues, Glass[,-10])
> # check if skewness is resolved
> skewValues2 <- apply(GlassBC, 2, skewness)
> skewValues2

         RI          Na          Mg          Al          Si           K
 1.56566039  0.03384644 -1.13645228  0.09105899 -0.65090568  6.46008890
         Ca          Ba          Fe
-0.19395573  3.36867997  1.72981071

>
```

Some variables are not transformed with BoxCox. I will try to center and scale them!

```
> normalValues <- preProcess(Glass[,-10], method = c("center", "scale"))
> normalValues

Created from 214 samples and 9 variables

Pre-processing:
  - centered (9)
  - ignored (0)
  - scaled (9)

> GlassNormal = predict(normalValues, Glass[,-10])
> skewValues3 = apply(GlassNormal,2,skewness)
> skewValues3

        RI          Na          Mg          Al          Si           K          Ca
 1.6027151  0.4478343 -1.1364523  0.8946104 -0.7202392  6.4600889  2.0184463
        Ba          Fe
 3.3686800  1.7298107
```

All 9 variables are scaled. But the skewness does not change. This makes sense actually. Normalizing the predictors does not change the shape.

I suspect some of these variables contain zeros. That's why log transformation is not possible.

```
> log = log(Glass[,c(3,6,8,9)])
> sum(Glass[,c(3,6,8,9)]==0)

[1] 392
```

Yes, lots of zeros.

I will try squared root on these transformatioin

```
> sqrt = sqrt(Glass[,c(3,6,8,9)])
> skewValues4 = apply(sqrt,2,skewness)
> skewValues4

        Mg          K         Ba         Fe
-1.3460193  0.8590459  2.3439793  1.0384996
```

Skewness is improved! I will use BoxCox on predictors without 0s, and squared root on predictors with 0s.

2 Exercise 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

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

  I use frequency ratio to detect degenerate distributioins and near-zero variance.

```
> library(mlbench)
> data(Soybean)
> distribution = apply(Soybean, 2, table)
> freqRatio = function(vector){
+   tab = table(vector)
+   tab.sort = sort(tab, TRUE)
+   return(tab.sort[1]/tab.sort[2])
+ }
> frequencyRatio = apply(Soybean, 2, freqRatio)
> sum(frequencyRatio > 20)

[1] 3
```

Yes, there are 3 variables that have frequency ratio above 20.

I can also use nearZeroVar to detect near zero variance predictors.

```
> nearZeroVar(Soybean)

[1] 19 26 28

> length(nearZeroVar(Soybean))

[1] 3
```

- 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?

  "it is important to know if the pattern of missing data is related to the outcome. This is called 'informative missingness' since the missing data pattern is instructional on its own" - Max Kuhn.

  I calculate proportion of missing values for each variable. I categorize those variables with more than 10% missing values as high NA group. Then, I create cross-tabulation between those high NA variable with class variable to see if there's structural information from the missing values.

```
> NAproportion = function(predictor){
+   #this function calculapredictorrtion of missing values within a variable
+   NAcount = sum(is.na(predictor))
+   return(NAcount/length(predictor))
+ }
> sort(apply(Soybean,2, NAproportion), TRUE)

          hail           sever        seed.tmt         lodging              ge
    0.177159590     0.177159590     0.177159590     0.177159590     0.1639824
      leaf.mild fruiting.bodies     fruit.spots   seed.discolor      shriveli
    0.158125915     0.155197657     0.155197657     0.155197657     0.1551976
    leaf.shread            seed     mold.growth       seed.size       leaf.ha
    0.146412884     0.134699854     0.134699854     0.134699854     0.1229868
      leaf.marg       leaf.size       leaf.malf      fruit.pods            prec
    0.122986823     0.122986823     0.122986823     0.122986823     0.0556368
   stem.cankers   canker.lesion       ext.decay        mycelium    int.discol
    0.055636896     0.055636896     0.055636896     0.055636896     0.0556368
      sclerotia     plant.stand           roots            temp       crop.hi
    0.055636896     0.052708638     0.045387994     0.043923865     0.0234260
   plant.growth            stem            date        area.dam             Cla
    0.023426061     0.023426061     0.001464129     0.001464129     0.0000000
         leaves
    0.000000000

> # find out which variables contain missing value above certain threshold
> highNAindex = which(apply(Soybean,2, NAproportion) > 0.10)
> highNAindex

          hail           sever        seed.tmt            germ       leaf.ha
             6               9              10              11
      leaf.marg       leaf.size     leaf.shread       leaf.malf       leaf.mi
            15              16              17              18
       lodging fruiting.bodies      fruit.pods     fruit.spots              se
            21              24              29              30
    mold.growth   seed.discolor       seed.size      shriveling
            32              33              34              35

>
```

From the high NA index, there are some predictors that have higher missing values. I will then look at how NAs are distributed within these predictors.

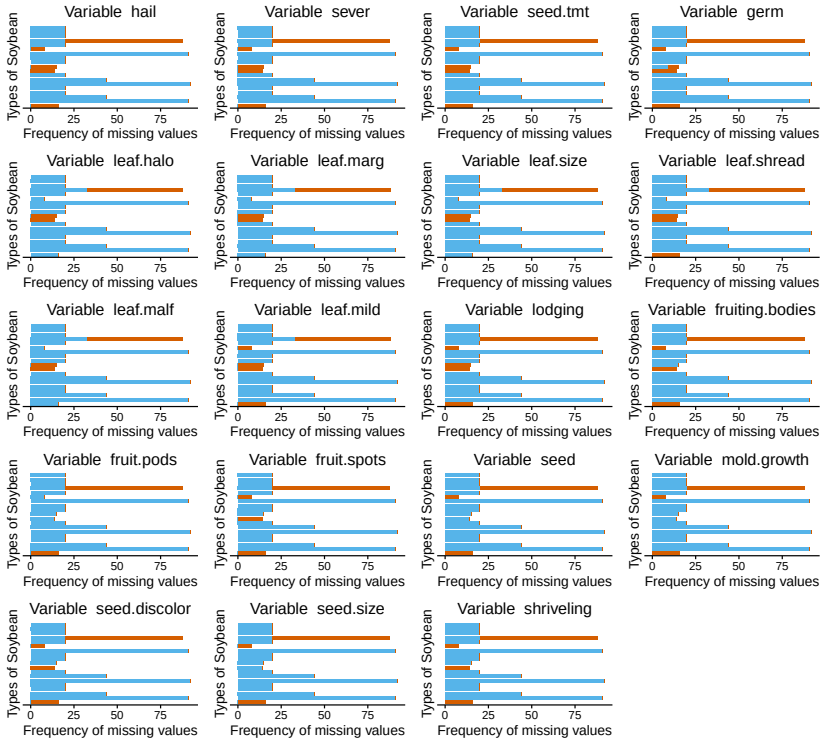
```
> # find out how the NAs are distributed
> crosstab = function(predictor){
+   # this function creates cross tabluation between a given variable and ou
+   tab = table(Soybean$Class, predictor, useNA = "always")
+   return(tab[-nrow(tab),])
+ }
> NApattern = apply(Soybean[,highNAindex], 2, crosstab)
```

Looking at the cross tabluation, there're a few classes that are associated with missing values.

I want to turn a table into a plot. I will plot each data frame as a horizontal stacked bar chart.

```
> table2stacked = function(NApattern,i){
+   var = names(NApattern[i])
+   table1 = NApattern[i][[1]]
+
+   df = as.data.frame(table1)
+   df[,"class"] = rownames(df)
+
+   # wide to long, because it's easier to plot
+
+   library(reshape2)
+   df3 = melt(df)
+   names(df3) = c("class","response", "frequency")
+   levels(df3$response)[!is.na(levels(df3$response))] <- "non-missing"
+   df3$response = addNA(df3$response)
+   levels(df3$response)[is.na(levels(df3$response))] <- "missing"
+
+
+   p = ggplot(df3, aes(x = class, y = frequency, fill = response))+
+         geom_bar(stat = "identity")+
+         scale_fill_manual(values=c("#56B4E9", "#D55E00", "grey"))+
+         coord_flip()+
```

```
+         labs(title=paste( "Variable ",var),
+               x = ("Types of Soybean"),
+               y = (paste("Frequency of missing values"))
+             )+
+         theme_classic()+
+         theme(legend.position="none")+
+         scale_x_discrete(breaks=NULL)
+
+
+   return(p)
+ }
> library(ggplot2)
> library(gridExtra)
> plot_list = list()
> length(NApattern)

[1] 19

> for(i in 1:19){
+   p = table2stacked(NApattern, i)
+   plot_list[[i]] = p
+   }
> grid.arrange(grobs = plot_list, ncol=4)
>
```



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

  In this dataset, the predictors with large proportion of missing values contain informative missingness. Therefore, I'd use imputation or tree-based method.

3 Exercise 3.3

Chapter 5 introduces Quantitative Structure-Activity Relationship (QSAR) modeling where the characteristics of a chemical compound are used to predict other chemical properties. The caret package contains a QSAR data set from Mente and Lombardo (2005). Here, the ability of a chemical to permeate the blood-brain barrier was experimentally determined for 208 compounds. 134 descriptors were measured for each compound.

- Start R and use these commands to load the data:

```
> library(caret)
> data(BloodBrain)
```

use ?BloodBrain to see more details The numeric outcome is contained in the vector logBBB while the predictors are in the data frame bbbDescr.

- Do any of the individual predictors have degenerate distributions?

```
> nearZeroVar(bbbDescr)

[1]  3 16 17 22 25 50 60

> length(nearZeroVar(bbbDescr))

[1] 7

>
```

Yes, there are 7 predictors that have frequency ratio below 0.05.

- 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?

```
> correlation = cor(bbbDescr)
> highCorr <- findCorrelation(correlation, cutoff = .75)
> length(highCorr)

[1] 66

> ncol(bbbDescr)

[1] 134
```

Yes, there are some that are highly correlated with each other. I can eliminate columns that are correlated with each above 0.75. This will remove 66 predictors, which amount of half the original data.