hw1

chris

2023-06-24

Christopher Huong SHG100

```
library(mlbench)
library(tidyverse)
library(psych)
library(caret)
library(naniar)
library(knitr)
data(Glass)

glimpse(Glass)
```

```
## Rows: 214
## Columns: 10
## $ RI   <dbl> 1.52101, 1.51761, 1.51618, 1.51766, 1.51742, 1.51596, 1.51743, 1.~
## $ Na   <dbl> 13.64, 13.89, 13.53, 13.21, 13.27, 12.79, 13.30, 13.15, 14.04, 13~
## $ Mg   <dbl> 4.49, 3.60, 3.55, 3.69, 3.62, 3.61, 3.60, 3.61, 3.58, 3.60, 3.46,~
## $ Al   <dbl> 1.10, 1.36, 1.54, 1.29, 1.24, 1.62, 1.14, 1.05, 1.37, 1.36, 1.56,~
## $ Si   <dbl> 71.78, 72.73, 72.99, 72.61, 73.08, 72.97, 73.09, 73.24, 72.08, 72~
## $ K    <dbl> 0.06, 0.48, 0.39, 0.57, 0.55, 0.64, 0.58, 0.57, 0.56, 0.57, 0.67,~
## $ Ca   <dbl> 8.75, 7.83, 7.78, 8.22, 8.07, 8.07, 8.17, 8.24, 8.30, 8.40, 8.09,~
## $ Ba   <dbl> 0, 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, 0.24,~
## $ Type <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
```

Exercise 3.1

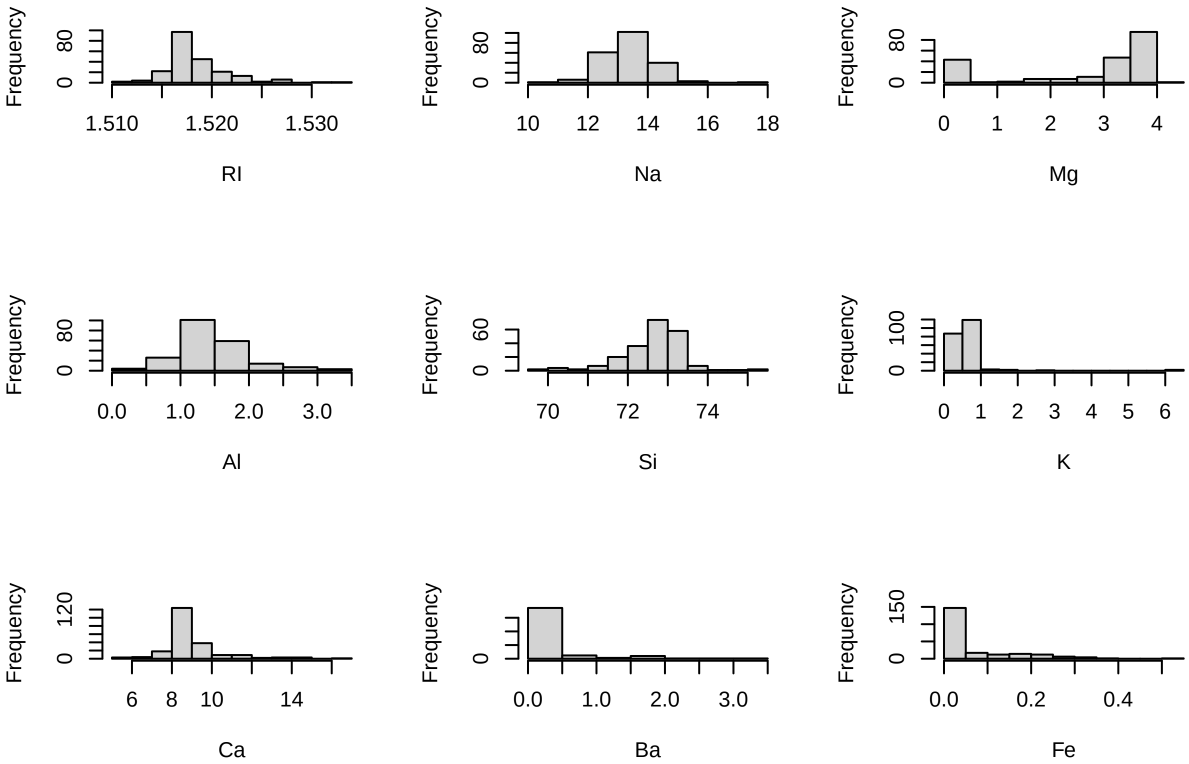
(a) Using visualizations, explore the predictor variables to understand their

distributions as well as the relationships between predictors.

Plot histograms of each predictor

```
vars_list <- as.list(colnames(select(Glass,-Type)))

par(mfrow=c(3,3))
for(i in vars_list){hist(select(Glass,-Type)[,i],xlab=i,main="")}
```



Based off histograms, Mg shows significant left skew, and K, Ca, Ba, and Fe show significant right skew.

Also, some predictors include 0, and some do not. Further, there is wide variability in the scale of the distribution (range of x-axis)

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

K, Ca, and Fe seem to have outliers at the far right of the distribution.

Compute skewness

```
describe(select(Glass, -Type))[, c(3,4,5,8,9,10,11)]
```

```
##     mean   sd median   min   max range  skew
## RI   1.52 0.00   1.52  1.51  1.53  0.02  1.60
## Na 13.41 0.82  13.30 10.73 17.38  6.65  0.45
## Mg   2.68 1.44   3.48  0.00  4.49  4.49 -1.14
## Al   1.44 0.50   1.36  0.29  3.50  3.21  0.89
## Si 72.65 0.77  72.79 69.81 75.41  5.60 -0.72
## K    0.50 0.65   0.56  0.00  6.21  6.21  6.46
## Ca   8.96 1.42   8.60  5.43 16.19 10.76  2.02
## Ba   0.18 0.50   0.00  0.00  3.15  3.15  3.37
## Fe   0.06 0.10   0.00  0.00  0.51  0.51  1.73
```

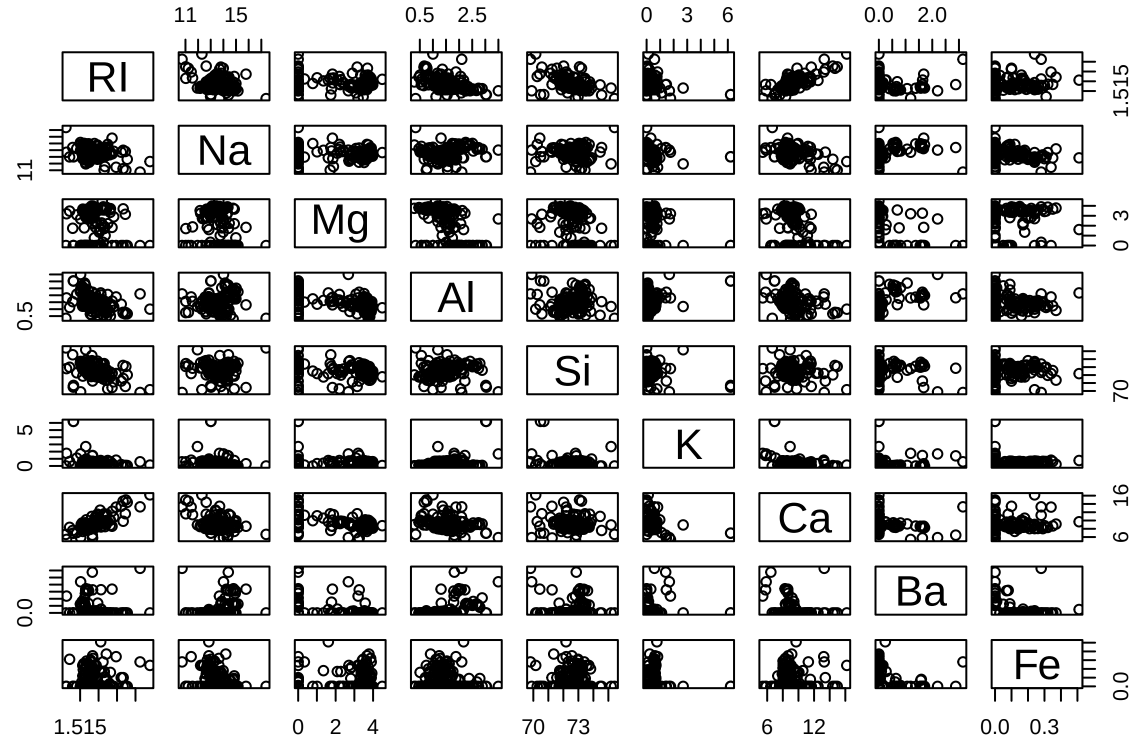
Skewness statistics show that Rl, K, Ca, Ba, and Fe are right skewed, and Mg is left skewed.

(c) Are there any relevant transformations of one or more predictors that

might improve the classification model?

Plot each predictor against each other

```
plot(select(Glass, -Type))
```



Visualizing pairwise scatterplots show that Rl and Ca are highly correlated.

Compute pairwise correlations between each predictor

```
cor(select(Glass, -Type), select(Glass, -Type))
```

```
##               RI          Na           Mg          Al          Si            K
## RI  1.0000000000 -0.19188538 -0.122274039 -0.40732603 -0.54205220 -0.289832711
## Na -0.1918853790  1.00000000 -0.273731961  0.15679367 -0.06980881 -0.266086504
## Mg -0.1222740393 -0.27373196  1.000000000 -0.48179851 -0.16592672  0.005395667
## Al -0.4073260341  0.15679367 -0.481798509  1.00000000 -0.00552372  0.325958446
## Si -0.5420521997 -0.06980881 -0.165926723 -0.00552372  1.00000000 -0.193330854
## K  -0.2898327111 -0.26608650  0.005395667  0.32595845 -0.19333085  1.000000000
## Ca  0.8104026963 -0.27544249 -0.443750026 -0.25959201 -0.20873215 -0.317836155
## Ba -0.0003860189  0.32660288 -0.492262118  0.47940390 -0.10215131 -0.042618059
## Fe  0.1430096093 -0.24134641  0.083059529 -0.07440215 -0.09420073 -0.007719049
##             Ca           Ba           Fe
## RI  0.8104027 -0.0003860189  0.143009609
## Na -0.2754425  0.3266028795 -0.241346411
## Mg -0.4437500 -0.4922621178  0.083059529
## Al -0.2595920  0.4794039017 -0.074402151
## Si -0.2087322 -0.1021513105 -0.094200731
## K  -0.3178362 -0.0426180594 -0.007719049
## Ca  1.0000000 -0.1128409671  0.124968219
## Ba -0.1128410  1.0000000000 -0.058691755
## Fe  0.1249682 -0.0586917554  1.000000000
```

Rl & Ca indeed show the highest correlation (r=0.81) and thus may be redundant as predictors in a model, and could be reduced or one predictor could be removed.

The other highly (above 0.5) correlated predictor is RI & Si (r=-0.54)

Perform PCA with mean centering and scaling. Compute variance with

```
pca <- prcomp(select(Glass, -Type),
 center = TRUE, scale. = TRUE)

summary(pca)
```

```
## Importance of components:
##                          PC1    PC2    PC3    PC4    PC5     PC6    PC7     PC8
## Standard deviation     1.585 1.4318 1.1853 1.0760 0.9560 0.72639 0.6074 0.25269
## Proportion of Variance 0.279 0.2278 0.1561 0.1286 0.1016 0.05863 0.0410 0.00709
## Cumulative Proportion  0.279 0.5068 0.6629 0.7915 0.8931 0.95173 0.9927 0.99982
##                            PC9
## Standard deviation     0.04011
## Proportion of Variance 0.00018
## Cumulative Proportion  1.00000
```

Reducing the data to 4 principal components retains 79% of the variance explained

Exercise 3.2

```
data("Soybean")
str(Soybean)
```

```
## 'data.frame':    683 obs. of  36 variables:
##  $ Class          : Factor w/ 19 levels "2-4-d-injury",..: 11 11 11 11 11 11 11 11 11 11 ...
##  $ date           : Factor w/ 7 levels "0","1","2","3",..: 7 5 4 4 7 6 6 5 7 5 ...
##  $ plant.stand    : Ord.factor w/ 2 levels "0"<"1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ precip         : Ord.factor w/ 3 levels "0"<"1"<"2": 3 3 3 3 3 3 3 3 3 3 ...
##  $ temp           : Ord.factor w/ 3 levels "0"<"1"<"2": 2 2 2 2 2 2 2 2 2 2 ...
##  $ hail           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ crop.hist      : Factor w/ 4 levels "0","1","2","3": 2 3 2 2 3 4 3 2 4 3 ...
##  $ area.dam       : Factor w/ 4 levels "0","1","2","3": 2 1 1 1 1 1 1 1 1 1 ...
##  $ sever          : Factor w/ 3 levels "0","1","2": 2 3 3 3 2 2 2 2 2 3 ...
##  $ seed.tmt       : Factor w/ 3 levels "0","1","2": 1 2 2 1 1 1 2 1 2 1 ...
##  $ germ           : Ord.factor w/ 3 levels "0"<"1"<"2": 1 2 3 2 3 2 1 3 2 3 ...
##  $ plant.growth   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ leaves         : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ leaf.halo      : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ leaf.marg      : Factor w/ 3 levels "0","1","2": 3 3 3 3 3 3 3 3 3 3 ...
##  $ leaf.size      : Ord.factor w/ 3 levels "0"<"1"<"2": 3 3 3 3 3 3 3 3 3 3 ...
##  $ leaf.shread    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ leaf.malf      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ leaf.mild      : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ stem           : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ lodging        : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 1 1 1 ...
##  $ stem.cankers   : Factor w/ 4 levels "0","1","2","3": 4 4 4 4 4 4 4 4 4 4 ...
##  $ canker.lesion  : Factor w/ 4 levels "0","1","2","3": 2 2 1 1 2 1 2 2 2 2 ...
##  $ fruiting.bodies: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ ext.decay      : Factor w/ 3 levels "0","1","2": 2 2 2 2 2 2 2 2 2 2 ...
##  $ mycelium       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ int.discolor   : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sclerotia      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ fruit.pods     : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ fruit.spots    : Factor w/ 4 levels "0","1","2","4": 4 4 4 4 4 4 4 4 4 4 ...
##  $ seed           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ mold.growth    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ seed.discolor  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ seed.size      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ shriveling     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ roots          : Factor w/ 3 levels "0","1","2": 1 1 1 1 1 1 1 1 1 1 ...
```

(a) Investigate the frequency distributions for the categorical predictors. Are

any of the distributions degenerate in the ways discussed earlier in this chapter?

```
describe(Soybean)[, c(3,4,5,8,9,10,11)]
```

```
##                  mean   sd median min max range  skew
## Class*           9.30 5.51      8   1  19    18  0.11
## date*            4.55 1.69      5   1   7     6 -0.30
## plant.stand*     1.45 0.50      1   1   2     1  0.19
## precip*          2.60 0.69      3   1   3     2 -1.42
## temp*            2.18 0.63      2   1   3     2 -0.16
## hail*            1.23 0.42      1   1   2     1  1.31
## crop.hist*       2.88 0.98      3   1   4     3 -0.40
## area.dam*        2.58 1.07      2   1   4     3  0.02
## sever*           1.73 0.60      2   1   3     2  0.17
## seed.tmt*        1.52 0.61      1   1   3     2  0.74
## germ*            2.05 0.79      2   1   3     2 -0.09
## plant.growth*    1.34 0.47      1   1   2     1  0.68
## leaves*          1.89 0.32      2   1   2     1 -2.44
## leaf.halo*       2.20 0.95      3   1   3     2 -0.41
## leaf.marg*       1.77 0.96      1   1   3     2  0.46
## leaf.size*       2.28 0.61      2   1   3     2 -0.25
## leaf.shread*     1.16 0.37      1   1   2     1  1.80
## leaf.malf*       1.08 0.26      1   1   2     1  3.22
## leaf.mild*       1.10 0.40      1   1   3     2  3.95
## stem*            1.56 0.50      2   1   2     1 -0.23
## lodging*         1.07 0.26      1   1   2     1  3.23
## stem.cankers*    2.06 1.35      1   1   4     3  0.61
## canker.lesion*   1.98 1.08      2   1   4     3  0.51
## fruiting.bodies* 1.18 0.38      1   1   2     1  1.66
## ext.decay*       1.25 0.48      1   1   3     2  1.70
## mycelium*        1.01 0.10      1   1   2     1 10.20
## int.discolor*    1.13 0.42      1   1   3     2  3.34
## sclerotia*       1.03 0.17      1   1   2     1  5.40
## fruit.pods*      1.50 0.88      1   1   4     3  1.84
## fruit.spots*     1.85 1.17      1   1   4     3  0.95
## seed*            1.19 0.40      1   1   2     1  1.54
## mold.growth*     1.11 0.32      1   1   2     1  2.43
## seed.discolor*   1.11 0.31      1   1   2     1  2.47
## seed.size*       1.10 0.30      1   1   2     1  2.66
## shriveling*      1.07 0.25      1   1   2     1  3.49
## roots*           1.18 0.44      1   1   3     2  2.46
```

Left skewed: precip, leaves Right skewed: hail, leaf.shread, leaf.malf, leaf.mild, lodging, fruiting.bodies, ext.decay, mycelium, int.discolor, sclerotia, fruit.pods, seed, mold.growth, seed.discolor, seed.size, shriveling, 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?

```
percentmiss <- function(x){
  sum(is.na(x)) / length(x) * 100}

apply(Soybean, 2, percentmiss)   ####percent missingness per col
```

```
##           Class            date     plant.stand          precip            temp
##       0.0000000       0.1464129       5.2708638       5.5636896       4.3923865
##            hail       crop.hist        area.dam           sever        seed.tmt
##      17.7159590       2.3426061       0.1464129      17.7159590      17.7159590
##            germ    plant.growth          leaves       leaf.halo       leaf.marg
##      16.3982430       2.3426061       0.0000000      12.2986823      12.2986823
##       leaf.size     leaf.shread       leaf.malf       leaf.mild            stem
##      12.2986823      14.6412884      12.2986823      15.8125915       2.3426061
##         lodging    stem.cankers   canker.lesion fruiting.bodies       ext.decay
##      17.7159590       5.5636896       5.5636896      15.5197657       5.5636896
##        mycelium    int.discolor       sclerotia      fruit.pods     fruit.spots
##       5.5636896       5.5636896       5.5636896      12.2986823      15.5197657
##            seed     mold.growth   seed.discolor       seed.size      shriveling
##      13.4699854      13.4699854      15.5197657      13.4699854      15.5197657
##           roots
##       4.5387994
```

```
#Class and leaves have no missing
```

Check percent of missing data by level of Class

```
missingbyclass <- apply(select(Soybean, -c(Class, leaves)), 2, function(x, y) {
  tab <- table(is.na(x), y)
  tab[2, ] / colSums(tab)
}, y = Soybean$Class)


missingbyclass <- missingbyclass[apply(missingbyclass, 1, sum) > 0,]
missingbyclass <- missingbyclass[, apply(missingbyclass, 2, sum) > 0]

t(missingbyclass)
```

```
##                 2-4-d-injury cyst-nematode diaporthe-pod-&-stem-blight
## date                  0.0625             0                         0.0
## plant.stand           1.0000             1                         0.4
## precip                1.0000             1                         0.0
## temp                  1.0000             1                         0.0
## hail                  1.0000             1                         1.0
## crop.hist             1.0000             0                         0.0
## area.dam              0.0625             0                         0.0
## sever                 1.0000             1                         1.0
## seed.tmt              1.0000             1                         1.0
## germ                  1.0000             1                         0.4
## plant.growth          1.0000             0                         0.0
## leaf.halo             0.0000             1                         1.0
## leaf.marg             0.0000             1                         1.0
## leaf.size             0.0000             1                         1.0
## leaf.shread           1.0000             1                         1.0
## leaf.malf             0.0000             1                         1.0
## leaf.mild             1.0000             1                         1.0
## stem                  1.0000             0                         0.0
## lodging               1.0000             1                         1.0
## stem.cankers          1.0000             1                         0.0
## canker.lesion         1.0000             1                         0.0
## fruiting.bodies       1.0000             1                         0.0
## ext.decay             1.0000             1                         0.0
## mycelium              1.0000             1                         0.0
## int.discolor          1.0000             1                         0.0
## sclerotia             1.0000             1                         0.0
## fruit.pods            1.0000             0                         0.0
## fruit.spots           1.0000             1                         0.0
## seed                  1.0000             0                         0.0
## mold.growth           1.0000             0                         0.0
## seed.discolor         1.0000             1                         0.0
## seed.size             1.0000             0                         0.0
## shriveling            1.0000             1                         0.0
## roots                 1.0000             0                         1.0
##                 herbicide-injury phytophthora-rot
## date                           0        0.0000000
## plant.stand                    0        0.0000000
## precip                         1        0.0000000
## temp                           0        0.0000000
## hail                           1        0.7727273
## crop.hist                      0        0.0000000
## area.dam                       0        0.0000000
## sever                          1        0.7727273
## seed.tmt                       1        0.7727273
## germ                           1        0.7727273
## plant.growth                   0        0.0000000
## leaf.halo                      0        0.6250000
## leaf.marg                      0        0.6250000
## leaf.size                      0        0.6250000
## leaf.shread                    0        0.6250000
## leaf.malf                      0        0.6250000
## leaf.mild                      1        0.6250000
## stem                           0        0.0000000
## lodging                        1        0.7727273
## stem.cankers                   1        0.0000000
## canker.lesion                  1        0.0000000
## fruiting.bodies                1        0.7727273
## ext.decay                      1        0.0000000
## mycelium                       1        0.0000000
## int.discolor                   1        0.0000000
## sclerotia                      1        0.0000000
## fruit.pods                     0        0.7727273
## fruit.spots                    1        0.7727273
## seed                           1        0.7727273
## mold.growth                    1        0.7727273
## seed.discolor                  1        0.7727273
## seed.size                      1        0.7727273
## shriveling                     1        0.7727273
## roots                          0        0.0000000
```

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

predictors or imputation

Lets just impute

```
library(mice)

# Soybean_imp <- mice(Soybean, maxit=3,m=3, seed=333)
```

Then use Rubin's rule to pool model estimates across imputations.

4.1

Note: For Exercise 4.1 (a) of your textbook, you just need to make comments based on Fig. 1.1 on page 7 of textbook. Since the data link is not available anymore, you do not need to use R code to access the data.

(a) What data splitting method(s) would you use for these data? Explain.

As the sample is large (N=12,495) you can split the sample to train the model, and test the model. If the uneven distribution of Genres is worrying, then using k-fold cross-validation can help ensure that each Genre is adequately sampled and not underrepresented in the training set due to chance.

4.4. Brodnjak-Vonina et al. (2005) develop a methodology for food laboratories to determine the type of oil from a sample. In their procedure, they used a gas chromatograph (an instrument that separate chemicals in a sample) to measure seven different fatty acids in an oil. These measurements would then be used to predict the type of oil in a food samples. To create their model, they used 96 samples2 of seven types of oils. These data can be found in the caret package using data(oil). The oil types are contained in a factor variable called oilType. The types are pumpkin(coded as A), sunflower (B), peanut (C), olive (D), soybean (E), rapeseed (F) and corn (G).

```
data(oil)
fattyAcids$oilType <- oilType
```

(a) Use the sample function in base R to create a completely random sample

of 60 oils. How closely do the frequencies of the random sample match the original samples? Repeat this procedure several times of understand the variation in the sampling process.

```
prop.table(table(fattyAcids$oilType))
```

```
##
##          A          B          C          D          E          F          G
## 0.38541667 0.27083333 0.03125000 0.07291667 0.11458333 0.10416667 0.02083333
```

```
set.seed(111)
sample1 <- fattyAcids[sample(1:nrow(fattyAcids), size = 60), ]
prop.table(table(sample1$oilType))
```

```
##
##          A          B          C          D          E          F          G
## 0.40000000 0.25000000 0.05000000 0.05000000 0.13333333 0.08333333 0.03333333
```

```
sample2 <- fattyAcids[sample(1:nrow(fattyAcids), size = 60), ]
prop.table(table(sample2$oilType))
```

```
##
##          A          B          C          D          E          F          G
## 0.36666667 0.31666667 0.03333333 0.05000000 0.11666667 0.10000000 0.01666667
```

```
sample3 <- fattyAcids[sample(1:nrow(fattyAcids), size = 60), ]
prop.table(table(sample3$oilType))
```

```
##
##          A          B          C          D          E          F          G
## 0.40000000 0.26666667 0.03333333 0.08333333 0.11666667 0.06666667 0.03333333
```

```
sample4 <- fattyAcids[sample(1:nrow(fattyAcids), size = 60), ]
prop.table(table(sample4$oilType))
```

```
##
##          A          B          C          D          E          F          G
## 0.38333333 0.23333333 0.05000000 0.10000000 0.11666667 0.08333333 0.03333333
```

Distributions of samples can vary relatively widely. For example C in the original sample is 3.1% of the data, while in sample 1 it is 5% of the data, which is a 161% larger.

(b) Use the caret package function createDataPartition to create a stratified

random sample. How does this compare to the completely random samples?

```
prop.table(table(fattyAcids$oilType))
```

```
##
##          A          B          C          D          E          F          G
## 0.38541667 0.27083333 0.03125000 0.07291667 0.11458333 0.10416667 0.02083333
```

```
set.seed(111)
strat <- createDataPartition(fattyAcids$oilType,
 p = .59,
 list= FALSE,
 times=4)

strat <- as.data.frame(strat)

strat_sample1 <- fattyAcids[strat$Resample1, ]
strat_sample2 <- fattyAcids[strat$Resample2, ]
strat_sample3 <- fattyAcids[strat$Resample3, ]
strat_sample4 <- fattyAcids[strat$Resample4, ]

prop.table(table(strat_sample1$oilType))
```

```
##
##          A          B          C          D          E          F          G
## 0.36666667 0.26666667 0.03333333 0.08333333 0.11666667 0.10000000 0.03333333
```

```
prop.table(table(strat_sample2$oilType))
```

```
##
##          A          B          C          D          E          F          G
## 0.36666667 0.26666667 0.03333333 0.08333333 0.11666667 0.10000000 0.03333333
```

```
prop.table(table(strat_sample3$oilType))
```

```
##
##          A          B          C          D          E          F          G
## 0.36666667 0.26666667 0.03333333 0.08333333 0.11666667 0.10000000 0.03333333
```

```
prop.table(table(strat_sample4$oilType))
```

```
##
##          A          B          C          D          E          F          G
## 0.36666667 0.26666667 0.03333333 0.08333333 0.11666667 0.10000000 0.03333333
```

The stratified samples are way closer (and more stable) to the original sample in oilType frequency

(c) With such a small samples size, what are the options for determining

performance of the model? Should a test set be used?

k-folds cross-validation can be used to test the performance of models built on small samples

(d) One method for understanding the uncertainty of a test set is to use a

confidence interval. To obtain a confidence interval for the overall accuracy, the based R function binom.test can be used. It requires the user to input the number of samples and the number correctly classified to calculate the interval. For example, suppose a test set sample of 20 oil samples was set aside and 76 were used for model training. For this test set size and a model that is about 80 % accurate (16 out of 20 correct), the confidence interval would be computed using binom.test(16, 20) Exact binomial test data: 16 and 20 number of successes = 16, number of trials = 20, p-value = 0.01182 alternative hypothesis: true probability of success is not equal to 0.5 95 percent confidence interval: 0.563386 0.942666 sample estimates: probability of success 0.8 In this case, the width of the 95 % confidence interval is 37.9 %. Try different samples sizes and accuracy rates to understand the trade-off between the uncertainty in the results, the model performance, and the test set size

```
binom.test(x=5, n=10)
```

```
##
##  Exact binomial test
##
## data:  5 and 10
## number of successes = 5, number of trials = 10, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.187086 0.812914
## sample estimates:
## probability of success
##                    0.5
```

```
binom.test(x=10, n=20)
```

```
##
##  Exact binomial test
##
## data:  10 and 20
## number of successes = 10, number of trials = 20, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.2719578 0.7280422
## sample estimates:
## probability of success
##                    0.5
```

```
binom.test(x=15, n=30)
```

```
##
##  Exact binomial test
##
## data:  15 and 30
## number of successes = 15, number of trials = 30, p-value = 1
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.3129703 0.6870297
## sample estimates:
## probability of success
##                    0.5
```

```
binom.test(x=30, n=40)
```

```
##
##  Exact binomial test
##
## data:  30 and 40
## number of successes = 30, number of trials = 40, p-value = 0.002221
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5880380 0.8730852
## sample estimates:
## probability of success
##                   0.75
```

The larger the test sample, the narrower the confidence interval (less uncertainty)

```
binom.test(x=5, n=40)
```

```
##
##  Exact binomial test
##
## data:  5 and 40
## number of successes = 5, number of trials = 40, p-value = 1.383e-06
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.04185963 0.26803292
## sample estimates:
## probability of success
##                  0.125
```

```
binom.test(x=10, n=40)
```

```
##
##  Exact binomial test
##
## data:  10 and 40
## number of successes = 10, number of trials = 40, p-value = 0.002221
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.1269148 0.4119620
## sample estimates:
## probability of success
##                   0.25
```

```
binom.test(x=15, n=40)
```

```
##
##  Exact binomial test
##
## data:  15 and 40
## number of successes = 15, number of trials = 40, p-value = 0.1539
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.2272627 0.5419852
## sample estimates:
## probability of success
##                  0.375
```

```
binom.test(x=30, n=40)
```

```
##
##  Exact binomial test
##
## data:  30 and 40
## number of successes = 30, number of trials = 40, p-value = 0.002221
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5880380 0.8730852
## sample estimates:
## probability of success
##                   0.75
```

The higher the accuracy, the narrower the confidence interval (less uncertainty)