

Homework 4

Applied Predictive Modeling - Data Pre-Processing & Over-Fitting and Model Tuning

Paul Perez

3/7/2021

Chapter 3 - Exercise 3.1

The UC Irvine Machine Learning Repository contains a data set related to glass identification. The data consists 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:

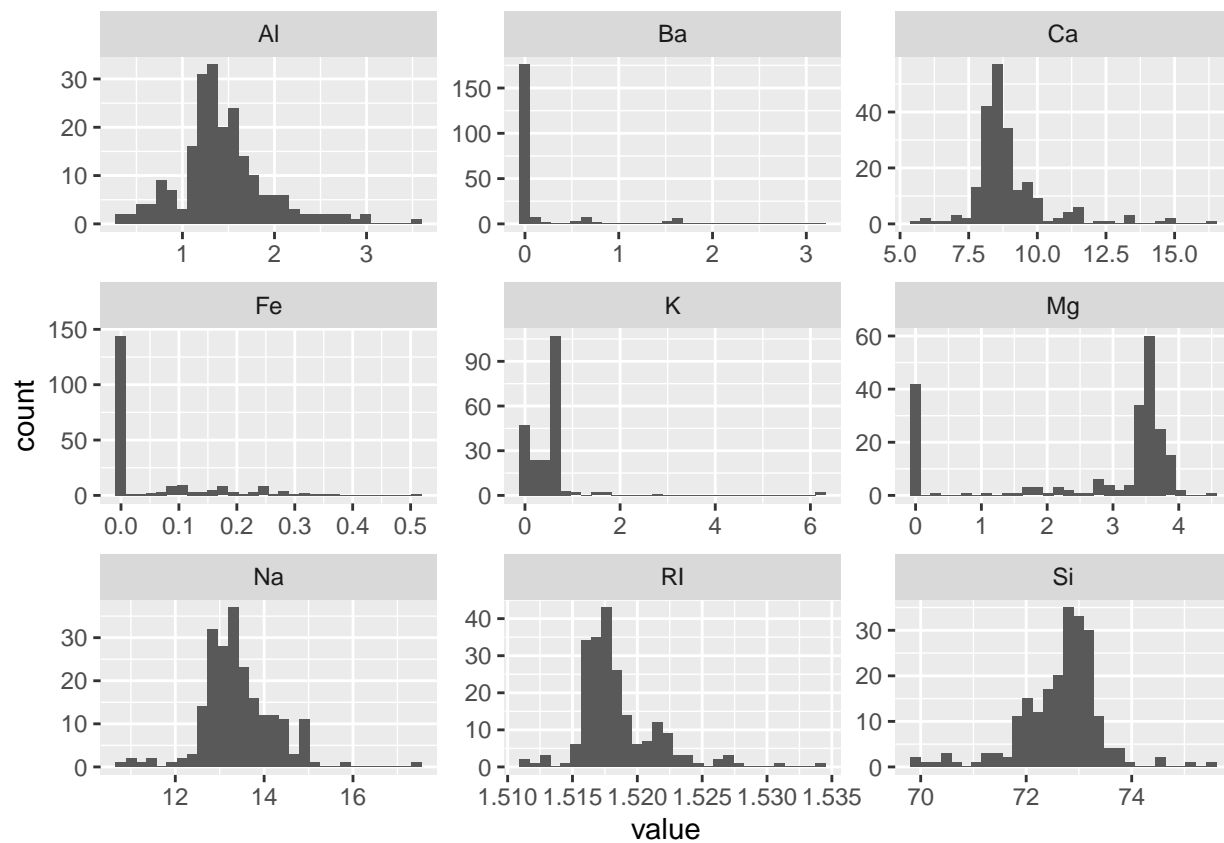
```
data(Glass)
str(Glass)

## 'data.frame':    214 obs. of  10 variables:
## $ RI : num  1.52 1.52 1.52 1.52 1.52 ...
## $ Na : num  13.6 13.9 13.5 13.2 13.3 ...
## $ Mg : num  4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...
## $ Al : num  1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...
## $ Si : num  71.8 72.7 73 72.6 73.1 ...
## $ K : num  0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...
## $ Ca : num  8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...
## $ Ba : num  0 0 0 0 0 0 0 0 0 0 ...
## $ Fe : num  0 0 0 0 0 0.26 0 0 0 0.11 ...
## $ Type: Factor w/ 6 levels "1","2","3","5",...: 1 1 1 1 1 1 1 1 1 1 ...
```

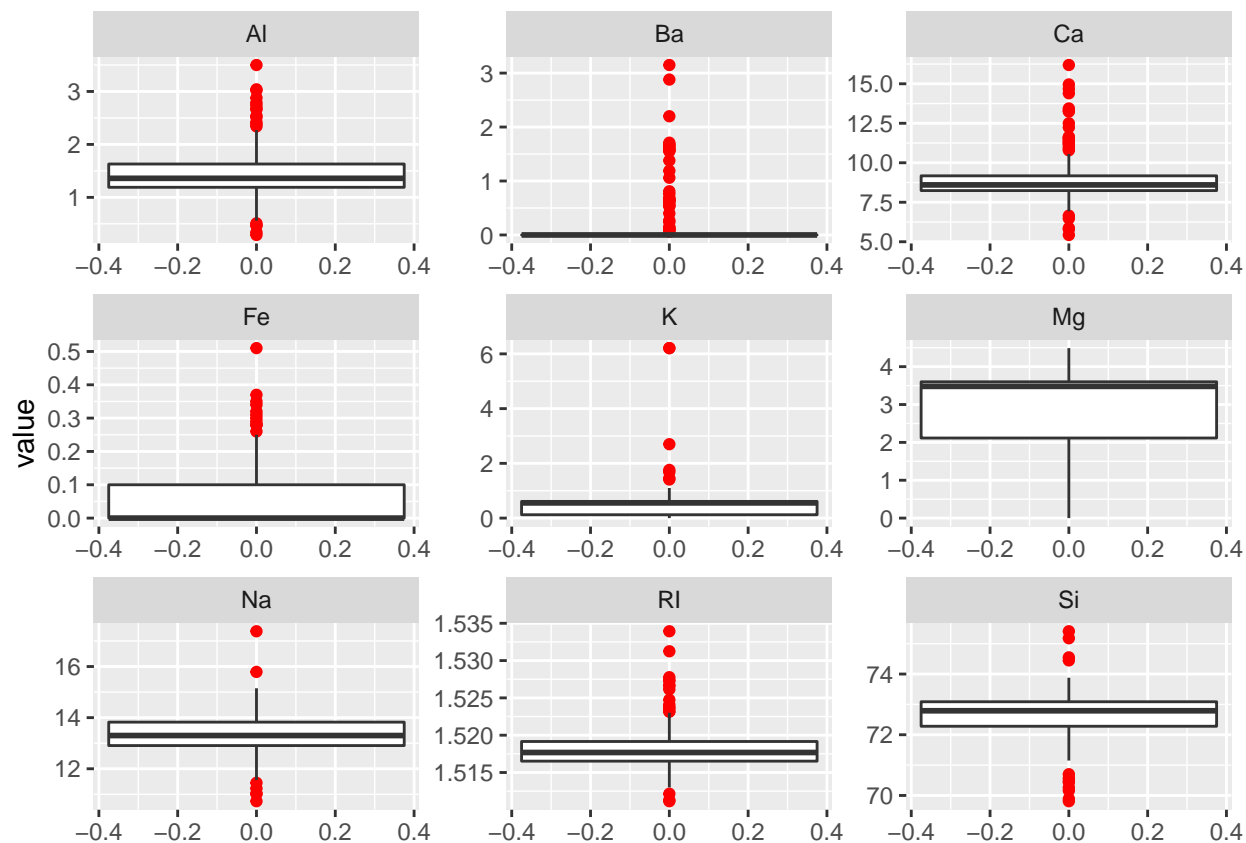
(a) Using visualizations, explore the predictor variables to understand their distributions as well as the relationships between predictors.

```
Glass %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram()
```

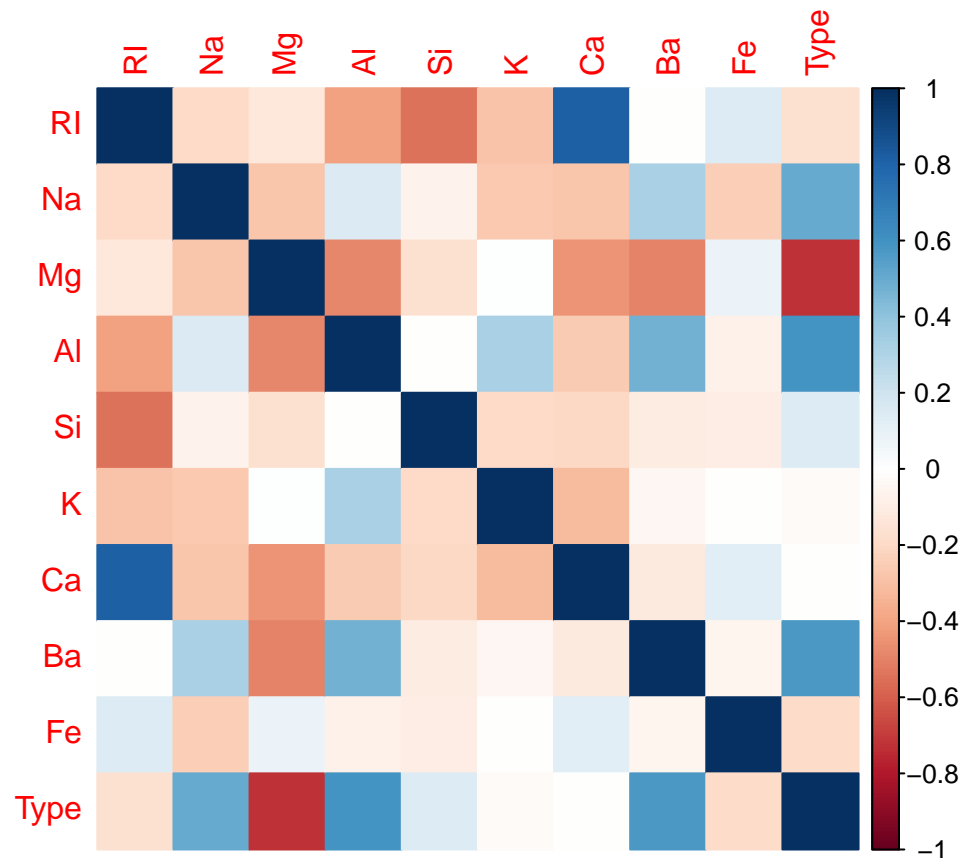
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
Glass %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_boxplot(outlier.color = "red") +
    coord_flip()
```



```
glass.matrix <- data.matrix(Glass, rownames.force = NA)
glass.corr <- cor(glass.matrix)
corrplot(glass.corr, method = "color")
```



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

While there are some outliers all of the predictor variables except Mg, one can argue that almost all of the predictors are skewed. Looking at the histograms above, we can see that predictor variables such as RI, Na, Al, and Si may show central distributions, they do appear to be slightly skewed.

Left-Skew Predictor Variables: - Mg - Si (More of symmetric distribution but shows slight left-skew)

Right-Skew Predictor Variables: - RI - Na (More of symmetric distribution but shows slight right-skew) - K - Ca - Ba - Fe

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

Two relevant transformation applicable here are both PCA and BoxCox transformation (both available in the **Caret** package). Given the boxplots above that show some outliers within our predictor variables, we can also apply the spacial sign transformation.

Chapter 3- 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. The data can be loaded via:

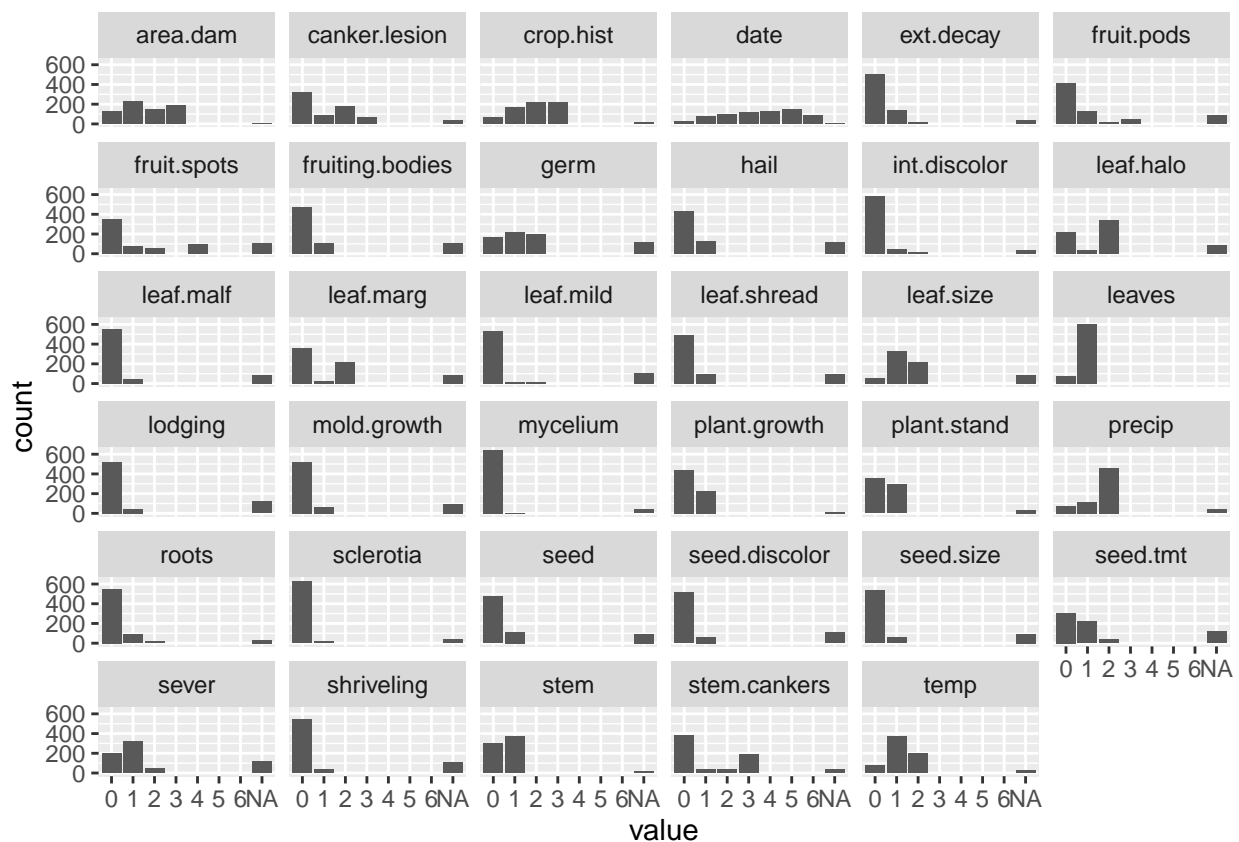
```
data(Soybean)
## See ?Soybean for details
str(Soybean)
```

```
## '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?

```
Soybean %>%
  select(-Class) %>%
  gather() %>%
  ggplot(aes(value)) +
    geom_bar() +
    facet_wrap(~ key)
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```



In the Computing section of Chapter 3 (3.8), we can filter our near zero variance columns using the `Caret` package to help identify any distributions that are degenerate.

```
nearZeroVar(Soybean)
```

```
## [1] 19 26 28
```

```
colnames(Soybean)[19]
```

```
## [1] "leaf.mild"
```

```
colnames(Soybean)[26]
```

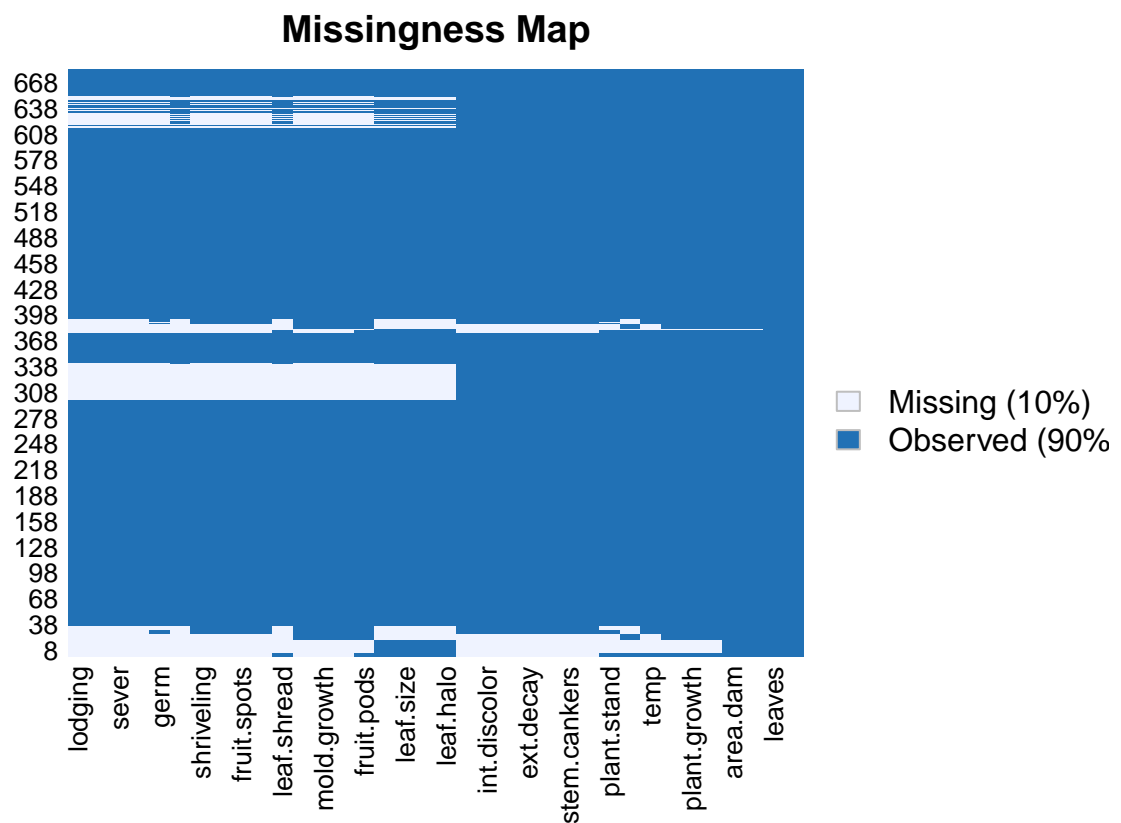
```
## [1] "mycelium"
```

```
colnames(Soybean)[28]
```

```
## [1] "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?

```
missmap(Soybean)
```



Looking at the missingnes map above from the `Amelia` package, we can see that those observations along the y-axis between the high 200's through the mid 300's show a pattern for missing.

(c) Develop a strategy for handling missing data, either by eliminating predictors or imputation.

The text describes using kNN modeling is one technique for imputing data. While my code keeps failing trying to apply a kNN Imputation, I would apply this after removing the near zero variance variables.

```
preprocess.strat <- function(df) {  
  
  for(i in nearZeroVar(df)) {  
    df <- df[-c(i)]  
  
  }  
  
  #df <- knn.impute(df)  
  return(df)  
}
```

```
soybean.impute <- preprocess.strat(Soybean)
```