Data 624 - HW4 (Fall 2024)

Khyati Naik

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(mlbench)
```

```
## Warning: package 'mlbench' was built under R version 4.3.2
```

```
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.

```
# Load necessary libraries
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
## Warning: package 'readr' was built under R version 4.3.2
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(caret)
```

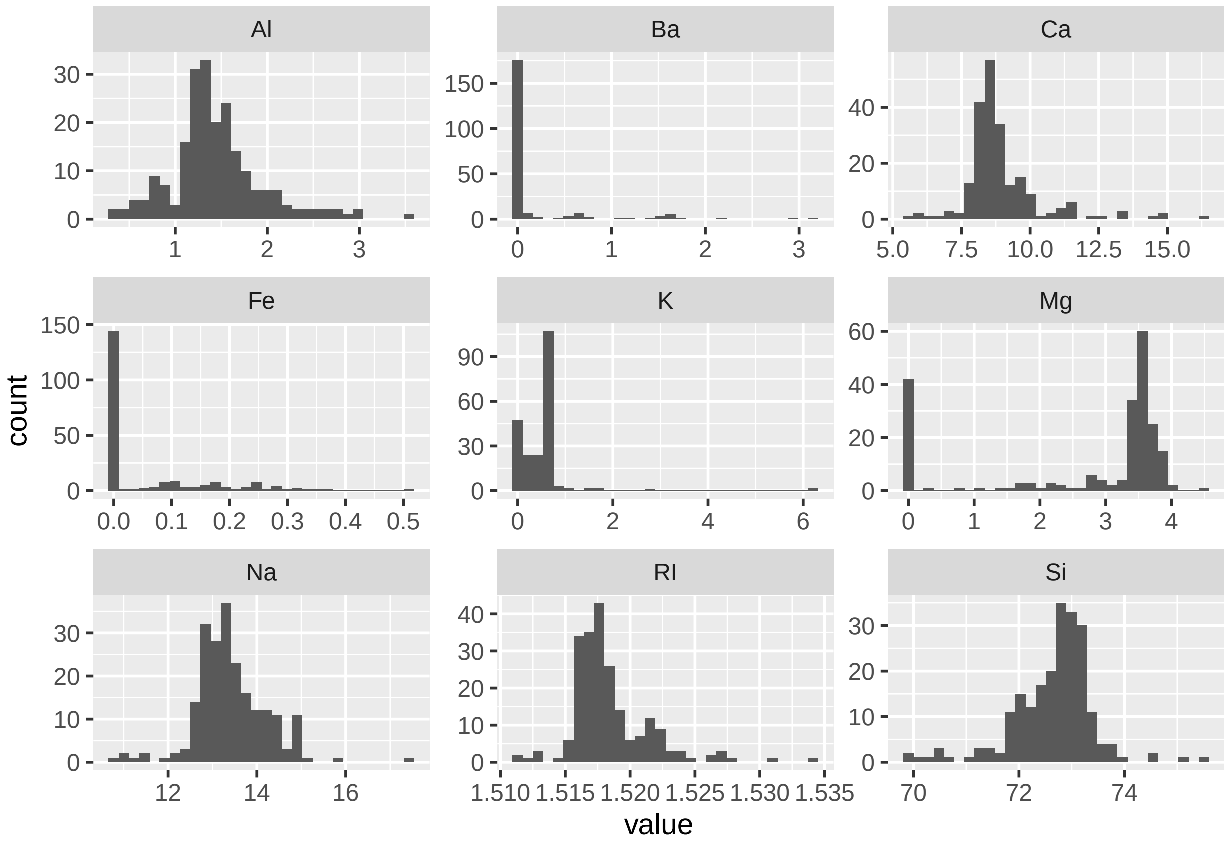
```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(mice)
```

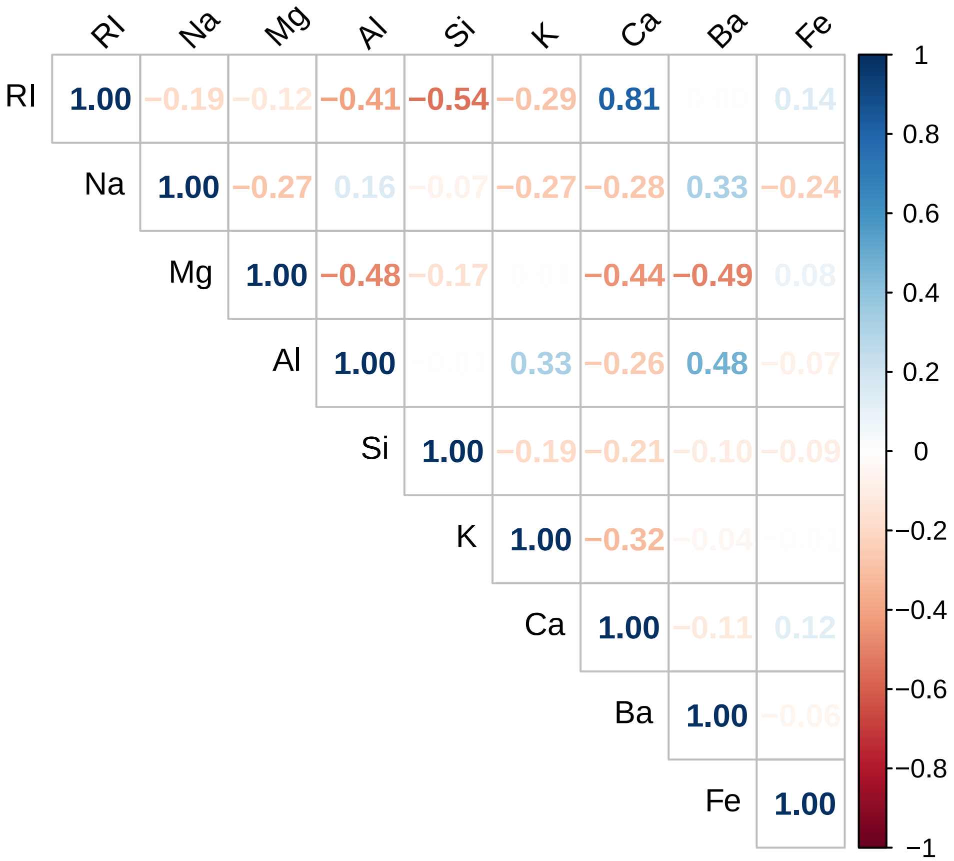
```
## Warning: package 'mice' was built under R version 4.3.3
```

```
##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```r
# Histograms for each predictor variable
Glass %>% select(-c(Type)) %>%
  gather() %>%
  ggplot(aes(x = value)) + geom_histogram(bins=30) + facet_wrap(~key, scales = 'free')
```



```r
# Correlation heatmap
cor_matrix <- cor(Glass[,1:9])
corrplot(cor_matrix, method = "number", type = "upper", tl.col = "black", tl.srt = 45)
```

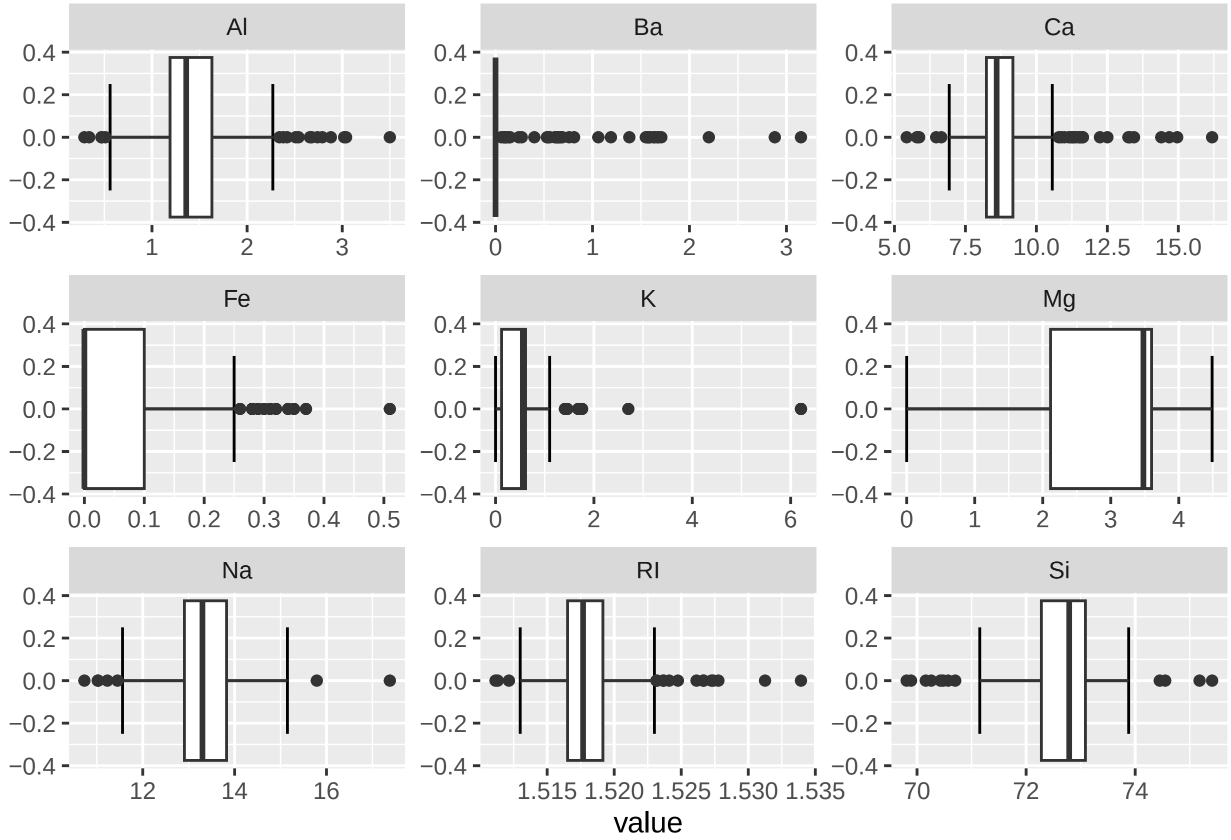


- The histograms for each predictor reveal notable differences in the distributions across variables. For instance, some predictors exhibit relatively normal distributions (e.g., Si and Ca), while others, such as Ba and K, show heavy skewness with a significant concentration of values near zero.
- The correlation heatmap provides valuable insight into the relationships between predictors. Strong positive correlations are observed between some variables (e.g., Ca and RI), indicating potential multicollinearity. However, others, such as Fe, show low correlations with most predictors, suggesting their independence. Understanding these correlations is important for model building, as highly correlated predictors can negatively impact some classification algorithms by introducing redundancy.

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

```r
# Boxplots for each predictor variable

Glass %>% select(-c(Type)) %>%
  gather() %>%
  ggplot(aes(x = value)) +
  stat_boxplot(geom = "errorbar", width = 0.5) +
  geom_boxplot() +
  facet_wrap(~key, scales = 'free')
```



- The boxplots provide clear evidence of outliers across several predictors. Notably, variables like Ba and K show extreme outliers, which could skew the classification model if not addressed.
- Skewness is evident in the distribution of several predictors, as seen in the histograms from the earlier visualization. For example, Ba and K exhibit strong right skewness, where most of the values are concentrated at lower levels, with a few high values pulling the tail to the right. These predictors may benefit from transformations to reduce skewness, making the data more suitable for classification algorithms that assume normality (e.g., linear discriminant analysis).

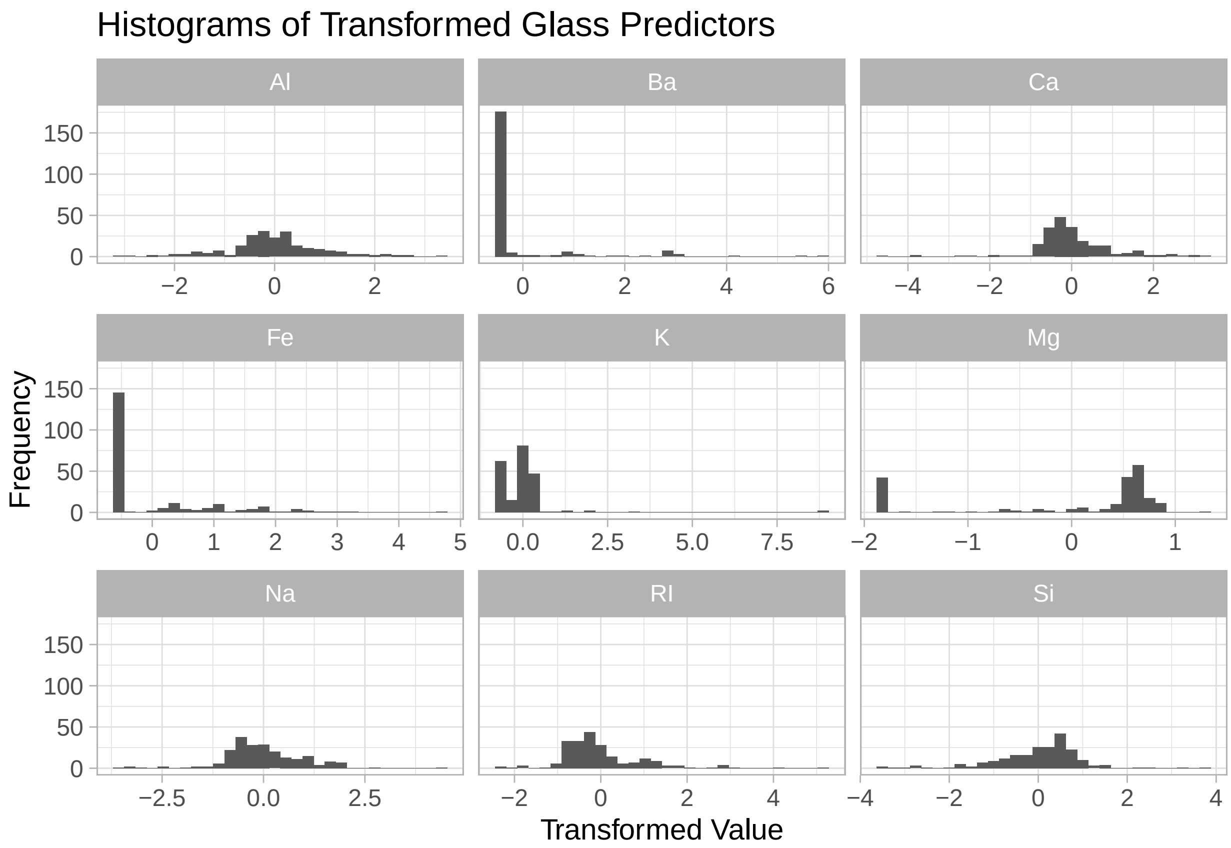
c. Are there any relevant transformations of one or more predictors that might improve the classification model?

```
# Perform Box-Cox transformation along with centering and scaling
transformation_model <- preProcess(Glass, method = c("BoxCox", "center", "scale"))

# Apply the transformations to the Glass dataset
Glass_modified <- predict(transformation_model, Glass)

# Exclude the target variable and reshape the data for visualization
Glass_modified %>%
  select(-Type) %>%
  pivot_longer(cols = everything()) %>%

  # Plot histograms of each transformed predictor
  ggplot(aes(x = value)) +
  geom_histogram(bins = 30) +
  facet_wrap(~name, scales = 'free_x') +
  theme_light() +
  labs(title = "Histograms of Transformed Glass Predictors", x = "Transformed Value", y = "Frequency")
```



- After applying the Box-Cox transformation, centering, and scaling, the distributions of predictors generally appear more normalized. The histograms of the transformed predictors show a reduction in skewness. By normalizing skewed data, the Box-Cox transformation enhances the applicability of machine learning models that are sensitive to the distribution of predictors.
- Centering and scaling also help standardize the predictors by ensuring that all variables have a mean of zero and a standard deviation of one. This step is particularly beneficial when using algorithms that rely on distance measures (e.g., k-nearest neighbors, support vector machines), as it ensures that all predictors contribute equally to the model.

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:

```
library(mlbench)
data(Soybean)
```

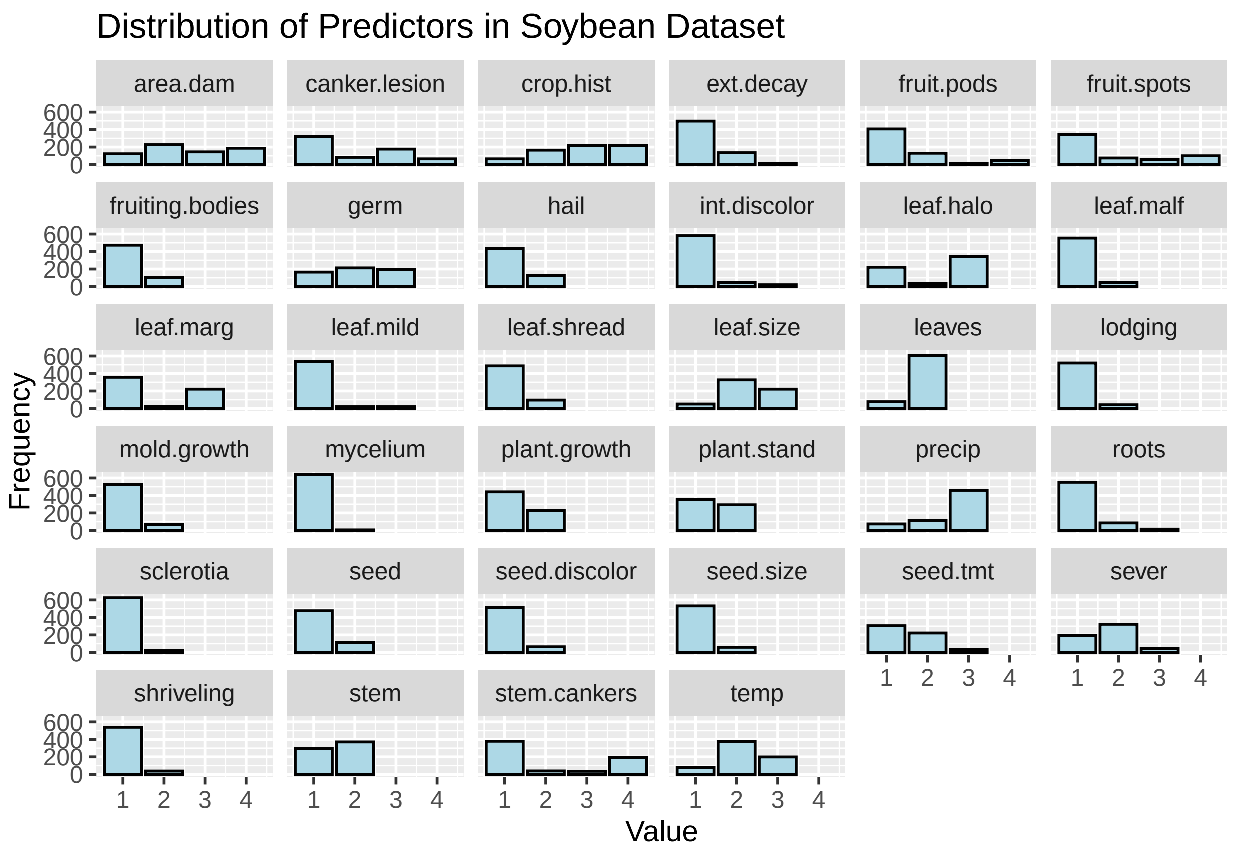
a. Investigate the frequency distributions for the categorical predictors. Are any of the distributions degenerate in the ways discussed earlier in this chapter?

```
# Transforming the Soybean dataset for analysis
Soybean_cleaned <- Soybean %>%
  select(-Class) %>%
  mutate(across(where(is.factor), as.numeric)) %>%
  pivot_longer(cols = -c(date), names_to = "Variable", values_to = "Value")

# Distribution of Predictors
ggplot(Soybean_cleaned, aes(x = Value)) +
  geom_histogram(fill = "lightblue", color = "black", stat="count") +
  facet_wrap(vars(Variable)) +
  labs(title = "Distribution of Predictors in Soybean Dataset",
       x = "Value",
       y = "Frequency")
```

```
## Warning in geom_histogram(fill = "lightblue", color = "black", stat = "count"):
## Ignoring unknown parameters: `binwidth`, `bins`, and `pad`
```

```
## Warning: Removed 2336 rows containing non-finite outside the scale range
## (`stat_count()`).
```



```r
# Summarize the categorical predictors
summary(Soybean)
```

```
##                  Class          date     plant.stand  precip      temp
##  brown-spot         : 92   5      :149   0   :354    0   : 74   0   : 80
##  alternarialeaf-spot: 91   4      :131   1   :293    1   :112   1   :374
##  frog-eye-leaf-spot : 91   3      :118   NA's: 36    2   :459   2   :199
##  phytophthora-rot   : 88   2      : 93               NA's: 38   NA's: 30
##  anthracnose        : 44   6      : 90
##  brown-stem-rot     : 44   (Other):101
##  (Other)            :233   NA's   :  1
##    hail     crop.hist  area.dam    sever     seed.tmt     germ     plant.growth
##  0   :435   0   : 65   0   :123   0   :195   0   :305   0   :165   0   :441
##  1   :127   1   :165   1   :227   1   :322   1   :222   1   :213   1   :226
##  NA's:121   2   :219   2   :145   2   : 45   2   : 35   2   :193   NA's: 16
##             3   :218   3   :187   NA's:121   NA's:121   NA's:112
##             NA's: 16   NA's:  1
##
##
##  leaves  leaf.halo  leaf.marg  leaf.size  leaf.shread leaf.malf  leaf.mild
##  0: 77   0   :221   0   :357   0   : 51   0   :487    0   :554   0   :535
##  1:606   1   : 36   1   : 21   1   :327   1   : 96    1   : 45   1   : 20
##          2   :342   2   :221   2   :221   NA's:100    NA's: 84   2   : 20
##          NA's: 84   NA's: 84   NA's: 84                          NA's:108
##
```

```
##
##
##    stem     lodging    stem.cankers canker.lesion fruiting.bodies ext.decay
##  0   :296   0   :520   0   :379     0   :320      0   :473        0   :497
##  1   :371   1   : 42   1   : 39     1   : 83      1   :104        1   :135
##  NA's: 16   NA's:121   2   : 36     2   :177      NA's:106        2   : 13
##                        3   :191     3   : 65                      NA's: 38
##                        NA's: 38     NA's: 38
##
##
##  mycelium   int.discolor sclerotia  fruit.pods fruit.spots   seed
##  0   :639   0   :581     0   :625   0   :407   0   :345    0   :476
##  1   :  6   1   : 44     1   : 20   1   :130   1   : 75    1   :115
##  NA's: 38   2   : 20     NA's: 38   2   : 14   2   : 57    NA's: 92
##             NA's: 38                3   : 48   4   :100
##                                     NA's: 84   NA's:106
##
##
##  mold.growth seed.discolor seed.size  shriveling  roots
##  0   :524    0   :513      0   :532   0   :539   0   :551
##  1   : 67    1   : 64      1   : 59   1   : 38   1   : 86
##  NA's: 92    NA's:106      NA's: 92   NA's:106   2   : 15
##                                                  NA's: 31
##
##
##
```

- The bar plots generated for the categorical predictors provide a clear visual overview of the distributions of each variable. While most predictors appear to be well-distributed, some may exhibit degenerate distributions where a large proportion of the observations fall into a single category. This can be seen in predictors such as precipitation or leaf spots, where the majority of the data might be clustered into one or two levels.
- Degenerate distributions, where one category overwhelmingly dominates, can reduce the predictive power of the model as they offer little variance to distinguish between different classes. This may warrant either the removal of such variables or further investigation to combine similar categories.
- The summary statistics also provide a useful overview, confirming that many variables contain multiple levels, but only a few may dominate the distribution in each case.

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?

```
# Count the number of observations with a missing value by predictor variable
colSums_Missing_Count <- data.frame(colSums(is.na(Soybean)))

# Name the column for the NA count
colnames(colSums_Missing_Count) <- "NA.Count"

# Convert the index column into a named column to keep the variable names
colSums_Missing_Count <- cbind(Variable = rownames(colSums_Missing_Count), colSums_Missing_Count)
rownames(colSums_Missing_Count) <- 1:nrow(colSums_Missing_Count)

# Sort by the missing count in descending order
```

```r
colSums_Missing_Count <- colSums_Missing_Count[order(-colSums_Missing_Count$NA.Count),]

# Output the results
(colSums_Missing_Count)
```

```
##           Variable NA.Count
## 6             hail      121
## 9            sever      121
## 10        seed.tmt      121
## 21         lodging      121
## 11            germ      112
## 19       leaf.mild      108
## 24 fruiting.bodies      106
## 30     fruit.spots      106
## 33   seed.discolor      106
## 35      shriveling      106
## 17     leaf.shread      100
## 31            seed       92
## 32     mold.growth       92
## 34       seed.size       92
## 14       leaf.halo       84
## 15       leaf.marg       84
## 16       leaf.size       84
## 18       leaf.malf       84
## 29      fruit.pods       84
## 4           precip       38
## 22    stem.cankers       38
## 23   canker.lesion       38
## 25       ext.decay       38
## 26        mycelium       38
## 27    int.discolor       38
## 28       sclerotia       38
## 3      plant.stand       36
## 36           roots       31
## 5             temp       30
## 7        crop.hist       16
## 12    plant.growth       16
## 20            stem       16
## 2             date        1
## 8         area.dam        1
## 1            Class        0
## 13          leaves        0
```

- The missing data analysis reveals that roughly 18% of the values in the dataset are missing, with certain predictors more affected than others. For example, the column-wise summary highlights predictors with a high percentage of missing values, such as plant growth or leaf conditions.
- From the analysis, predictors with a large amount of missing data could potentially be less reliable for modeling unless there is a pattern to the missingness that can be explained by other variables or related to specific class labels. If missing data is associated with specific outcome classes (e.g., missing leaf spot information in particular disease cases), this could provide useful insights but also lead to potential bias if not handled correctly.
- Overall, the missing data seems to be non-uniformly distributed across predictors, meaning a targeted approach for imputation or elimination is required.

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

```r
# Identify predictors with near-zero variance
nzv <- nearZeroVar(Soybean, saveMetrics = TRUE)
print(nzv)
```

```
##                  freqRatio percentUnique zeroVar   nzv
## Class             1.010989     2.7818448   FALSE FALSE
## date              1.137405     1.0248902   FALSE FALSE
## plant.stand       1.208191     0.2928258   FALSE FALSE
## precip            4.098214     0.4392387   FALSE FALSE
## temp              1.879397     0.4392387   FALSE FALSE
## hail              3.425197     0.2928258   FALSE FALSE
## crop.hist         1.004587     0.5856515   FALSE FALSE
## area.dam          1.213904     0.5856515   FALSE FALSE
## sever             1.651282     0.4392387   FALSE FALSE
## seed.tmt          1.373874     0.4392387   FALSE FALSE
## germ              1.103627     0.4392387   FALSE FALSE
## plant.growth      1.951327     0.2928258   FALSE FALSE
## leaves            7.870130     0.2928258   FALSE FALSE
## leaf.halo         1.547511     0.4392387   FALSE FALSE
## leaf.marg         1.615385     0.4392387   FALSE FALSE
## leaf.size         1.479638     0.4392387   FALSE FALSE
## leaf.shread       5.072917     0.2928258   FALSE FALSE
## leaf.malf        12.311111     0.2928258   FALSE FALSE
## leaf.mild        26.750000     0.4392387   FALSE  TRUE
## stem              1.253378     0.2928258   FALSE FALSE
## lodging          12.380952     0.2928258   FALSE FALSE
## stem.cankers      1.984293     0.5856515   FALSE FALSE
## canker.lesion     1.807910     0.5856515   FALSE FALSE
## fruiting.bodies   4.548077     0.2928258   FALSE FALSE
## ext.decay         3.681481     0.4392387   FALSE FALSE
## mycelium        106.500000     0.2928258   FALSE  TRUE
## int.discolor     13.204545     0.4392387   FALSE FALSE
## sclerotia        31.250000     0.2928258   FALSE  TRUE
## fruit.pods        3.130769     0.5856515   FALSE FALSE
## fruit.spots       3.450000     0.5856515   FALSE FALSE
## seed              4.139130     0.2928258   FALSE FALSE
## mold.growth       7.820896     0.2928258   FALSE FALSE
## seed.discolor     8.015625     0.2928258   FALSE FALSE
## seed.size         9.016949     0.2928258   FALSE FALSE
## shriveling       14.184211     0.2928258   FALSE FALSE
## roots             6.406977     0.4392387   FALSE FALSE
```

```r
# Remove near-zero variance predictors
Soybean_clean <- Soybean[, -nzv$nzv]

# Impute missing data using predictive mean matching for numeric variables
imputed_data <- mice(Soybean_clean, m = 5, maxit = 5, method = "norm.predict", seed = 500)
```

```
## Warning: Type mismatch for variable(s): date
## Imputation method norm.predict is not for factors with >2 levels.
## Warning: Type mismatch for variable(s): plant.stand
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): precip
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): temp
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): hail
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): crop.hist
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): area.dam
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): sever
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): seed.tmt
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): germ
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): plant.growth
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): leaf.halo
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): leaf.marg
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): leaf.size
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): leaf.shread
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): leaf.malf
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): leaf.mild
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): stem
## Imputation method norm.predict is not for factors.
## Warning: Type mismatch for variable(s): lodging
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): stem.cankers
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): canker.lesion
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): fruiting.bodies
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): ext.decay
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): mycelium
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): int.discolor
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): sclerotia
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): fruit.pods
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): fruit.spots
## Imputation method norm.predict is not for factors with >2 levels.

## Warning: Type mismatch for variable(s): seed
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): mold.growth
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): seed.discolor
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): seed.size
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): shriveling
## Imputation method norm.predict is not for factors.

## Warning: Type mismatch for variable(s): roots
## Imputation method norm.predict is not for factors with >2 levels.

##
##  iter imp variable
##   1   1  date
## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

## Warning in `[<-.factor`(`*tmp*`, cc, value = structure(3.10283374779887, dim =
## c(1L, : invalid factor level, NA generated

##   plant.stand

## Warning in Ops.ordered(y, z$residuals): '-' is not meaningful for ordered
## factors

## Warning in Ops.ordered(y, z$residuals): invalid factor level, NA generated

##   precip

## Warning in Ops.ordered(y, z$residuals): '-' is not meaningful for ordered
## factors

## Warning in Ops.ordered(y, z$residuals): invalid factor level, NA generated

##   temp  hail

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

## *  crop.hist  area.dam  sever  seed.tmt  germ  plant.growth  leaf.halo  leaf.marg  leaf.size  leaf.s
##   1   2  date

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

## Warning in Ops.factor(y, z$residuals): invalid factor level, NA generated

##   plant.stand

## Warning in Ops.ordered(y, z$residuals): '-' is not meaningful for ordered
## factors

## Warning in Ops.ordered(y, z$residuals): invalid factor level, NA generated

##   precip

## Warning in Ops.ordered(y, z$residuals): '-' is not meaningful for ordered
## factors

## Warning in Ops.ordered(y, z$residuals): invalid factor level, NA generated

##   temp  hail

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

## *  crop.hist  area.dam  sever  seed.tmt  germ  plant.growth  leaf.halo  leaf.marg  leaf.size  leaf.s
##   1   3  date
## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

## Warning in Ops.factor(y, z$residuals): invalid factor level, NA generated

##   plant.stand

## Warning in Ops.ordered(y, z$residuals): '-' is not meaningful for ordered
## factors

## Warning in Ops.ordered(y, z$residuals): invalid factor level, NA generated

##   precip

## Warning in Ops.ordered(y, z$residuals): '-' is not meaningful for ordered
## factors

## Warning in Ops.ordered(y, z$residuals): invalid factor level, NA generated

##   temp  hail

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

## *  crop.hist  area.dam  sever  seed.tmt  germ  plant.growth  leaf.halo  leaf.marg  leaf.size  leaf.s
##   1   4  date

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

## Warning in Ops.factor(y, z$residuals): invalid factor level, NA generated

##   plant.stand

## Warning in Ops.ordered(y, z$residuals): '-' is not meaningful for ordered
## factors

## Warning in Ops.ordered(y, z$residuals): invalid factor level, NA generated

##   precip

## Warning in Ops.ordered(y, z$residuals): '-' is not meaningful for ordered
## factors

## Warning in Ops.ordered(y, z$residuals): invalid factor level, NA generated

##   temp  hail

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

## *  crop.hist  area.dam  sever  seed.tmt  germ  plant.growth  leaf.halo  leaf.marg  leaf.size  leaf.s
##   1   5  date
## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

## Warning in Ops.factor(y, z$residuals): invalid factor level, NA generated

##   plant.stand

## Warning in Ops.ordered(y, z$residuals): '-' is not meaningful for ordered
## factors

## Warning in Ops.ordered(y, z$residuals): invalid factor level, NA generated

##   precip

## Warning in Ops.ordered(y, z$residuals): '-' is not meaningful for ordered
## factors

## Warning in Ops.ordered(y, z$residuals): invalid factor level, NA generated

##   temp  hail

## Warning in Ops.factor(y, z$residuals): '-' not meaningful for factors

## *  crop.hist  area.dam  sever  seed.tmt  germ  plant.growth  leaf.halo  leaf.marg  leaf.size  leaf.s
##   2   1  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   2   2  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   2   3  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   2   4  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   2   5  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   3   1  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   3   2  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   3   3  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   3   4  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   3   5  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   4   1  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   4   2  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   4   3  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   4   4  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   4   5  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   5   1  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   5   2  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   5   3  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   5   4  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr
##   5   5  date  plant.stand  precip  temp  hail  crop.hist  area.dam  sever  seed.tmt  germ  plant.gr

## Warning: Number of logged events: 20
```

```r
Soybean_imputed <- complete(imputed_data, 1)
```

- The approach taken to handle missing data first identifies predictors with near-zero variance, removing variables that contribute little meaningful information. This step is crucial, as near-zero variance predictors can introduce noise and complexity into the model without providing any useful insights.

- For the remaining missing data, multiple imputation using the "norm.predict" method is applied. This is an effective strategy as it leverages the relationships between predictors to fill in missing values rather than simply dropping rows or using mean imputation. This maintains the dataset's integrity by preserving as much information as possible.
- The final dataset is checked for remaining missing values, confirming that imputation has been successfully applied. This approach balances the elimination of low-information predictors with a robust imputation strategy, ensuring that the cleaned dataset is suitable for predictive modeling.