# Data Preprocessing/Overfitting

Souleymane Doumbia

2024-09-30

## Contents



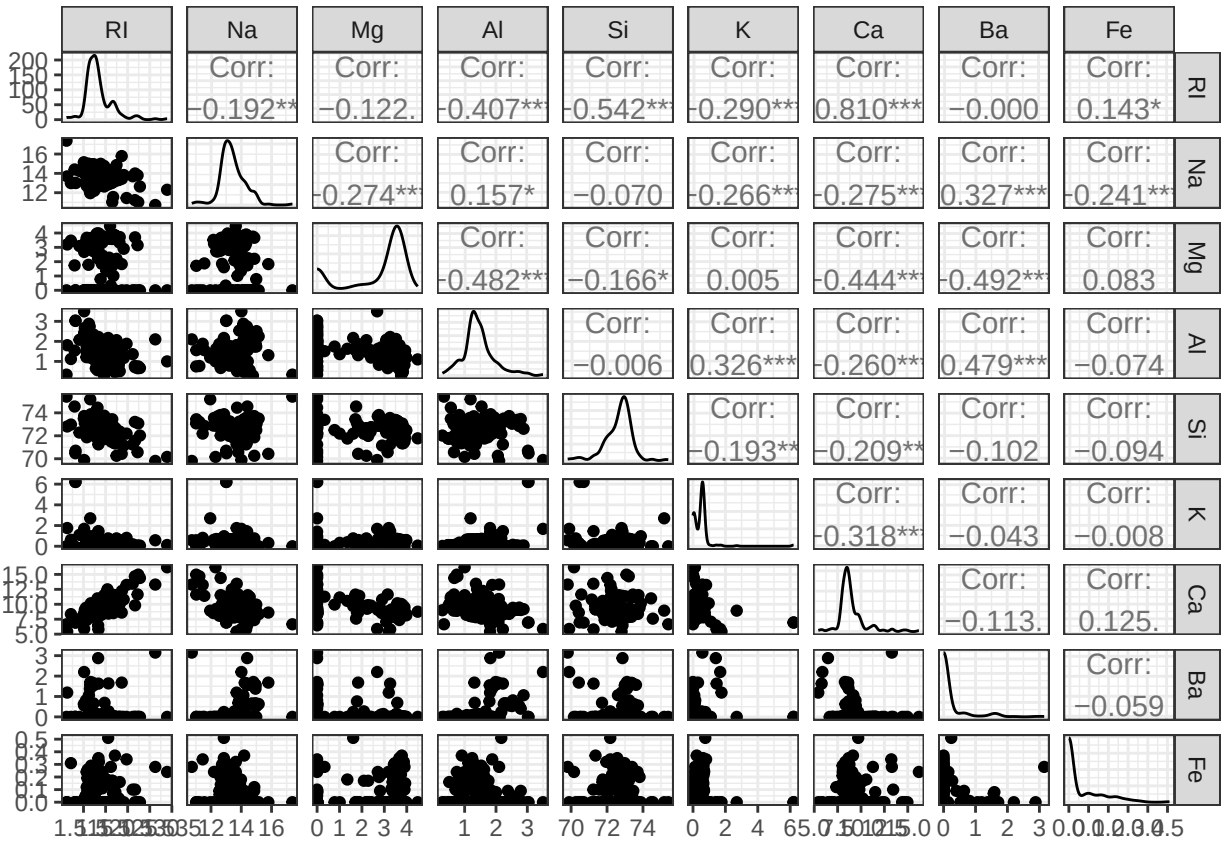
### Exercise 3.1: Exploring the Glass Dataset

#### (a) Using visualizations to explore the predictor variables

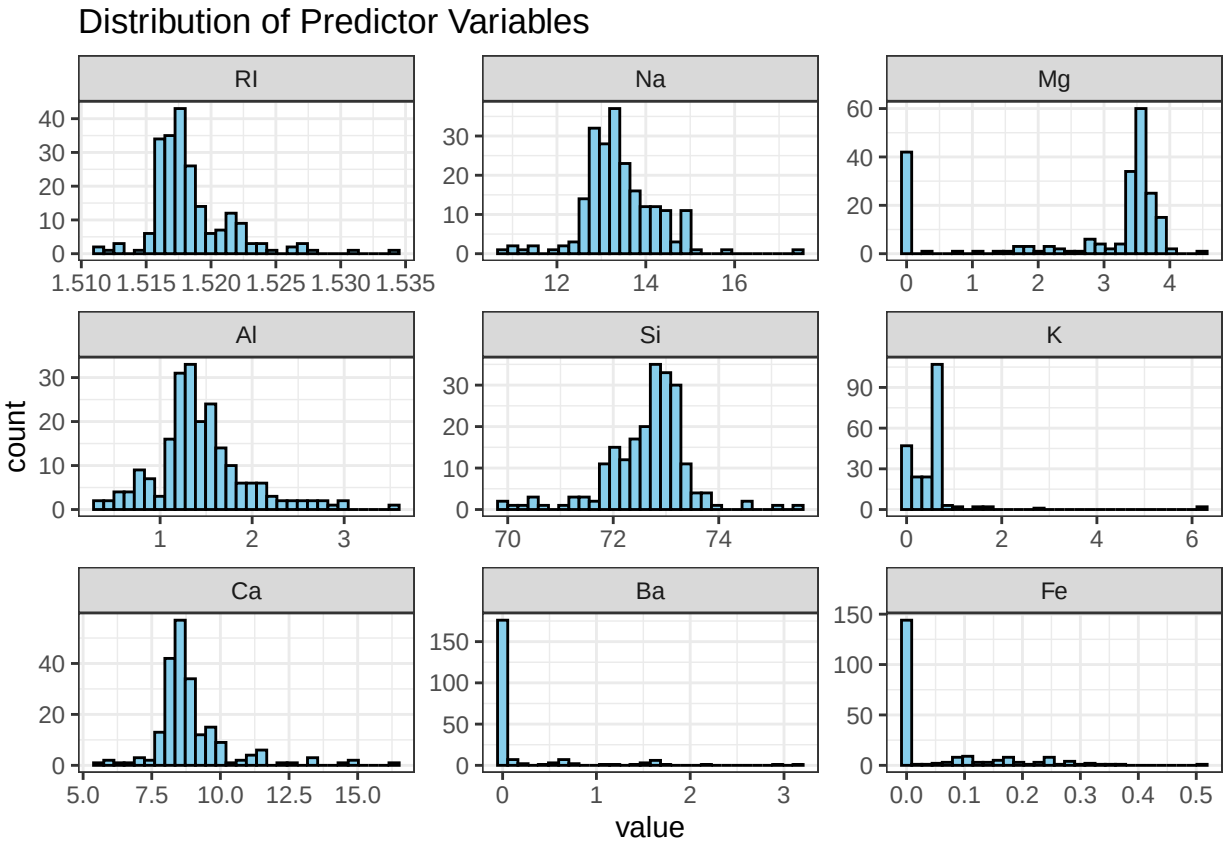
```
# Load the Glass data
data(Glass)
Glass <- Glass

# Basic structure of the dataset
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 ...
```

```
# Plotting pairwise relationships between predictors
# And excluding the Type variable as it's the target variable
ggpairs(Glass[, -10]) + theme_bw()
```




```
# Univariate plots for individual predictors
Glass_long <- reshape2::melt(Glass[, -10])
ggplot(Glass_long, aes(value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~variable, scales = "free") +
  theme_bw() +
  labs(title = "Distribution of Predictor Variables")
```

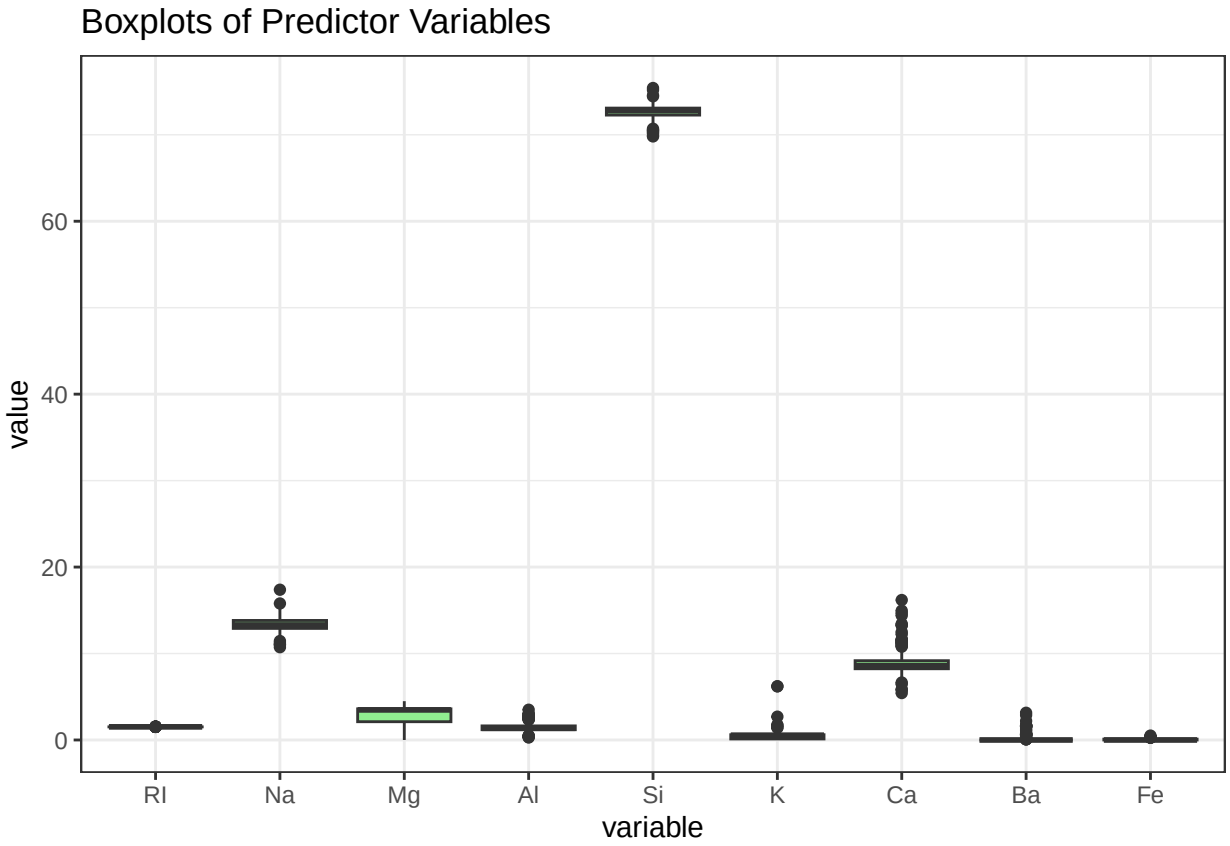



The visualizations of the predictor variables show the distributions and relationships between the nine predictors. From the univariate plots, several variables appear to follow relatively normal distributions, though some variables like *Mg*, *K*, *Ba*, and *Fe* display significant skewness, with most values concentrated towards the lower end.

The pair plot (scatter plot matrix) provides an overview of the relationships between these predictors. For example: - *Refractive Index (RI)* shows a slight negative correlation with *Na*, *Mg*, *Al*, *Si*, and *Ca*. - The variable *Ba* exhibits a positive correlation with *K* and *Ca*, which might indicate a link between these elements in certain types of glass.

### (b) Identifying outliers and skewness

```
# Boxplots to identify outliers
ggplot(Glass_long, aes(x = variable, y = value)) +
  geom_boxplot(fill = "lightgreen") +
  theme_bw() +
  labs(title = "Boxplots of Predictor Variables")
```




```
# Checking skewness for each predictor
library(e1071)
skewness_values <- apply(Glass[, -10], 2, skewness)
skewness_values
```

```
##         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
```

**Outliers**: Based on the boxplots, there are notable outliers in variables such as *Na*, *Mg*, *K*, *Ca*, and *Ba*. The distribution of these variables shows some extreme values that might represent specific glass types or experimental errors.

**Skewness**: Variables like *Mg*, *K*, *Ba*, and *Fe* are particularly skewed. For example: - *Mg* is negatively skewed with a large number of data points clustered at the high end (around 3.5–4.5) and very few observations in the lower range. - *K*, *Ba*, and *Fe* are positively skewed, with most values concentrated near zero and some extreme outliers at the higher end.

The presence of skewness suggests that transformations might be needed to improve the predictive modeling.

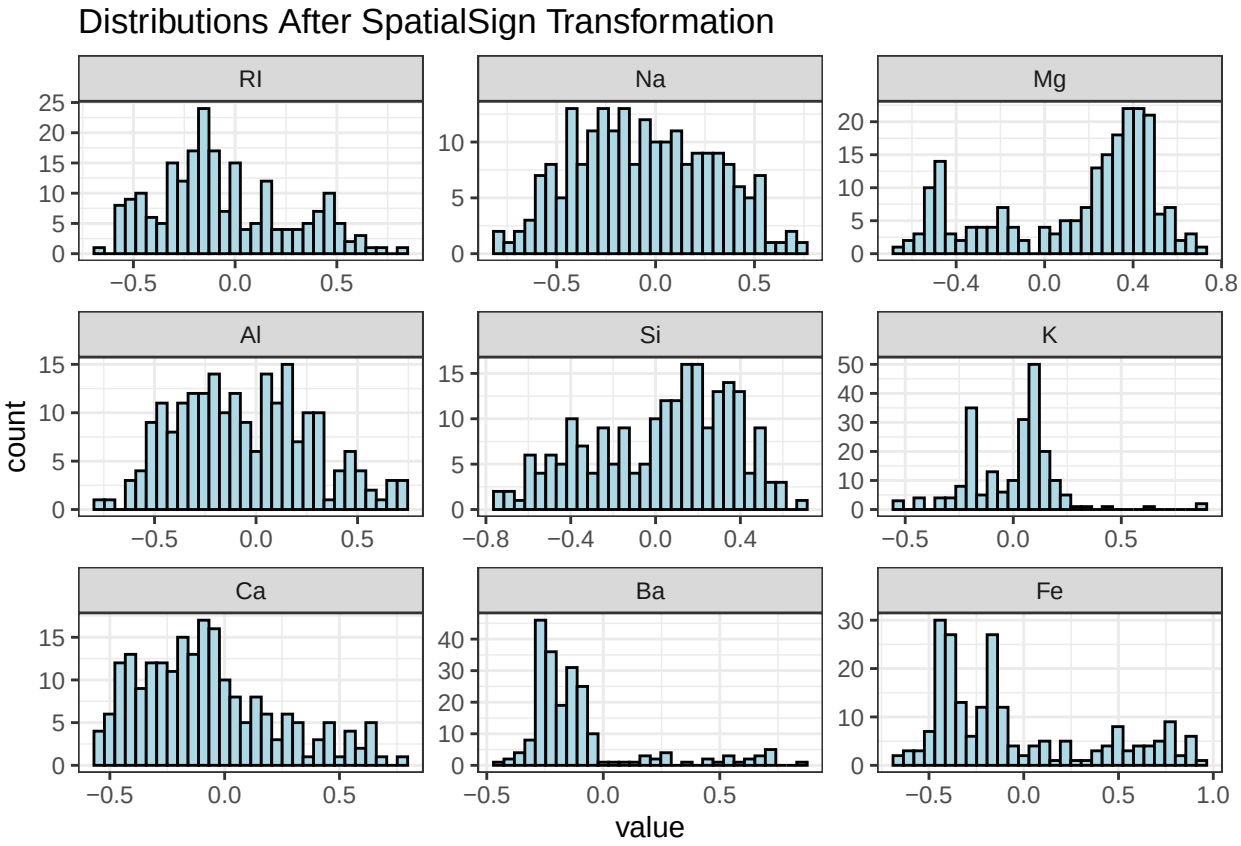
### (c) Transformations

```
# Loading necessary library
library(caret)

# Pre-process the Glass data using the 'spatialSign' method
preProc <- preProcess(Glass[, -10], method = "spatialSign")
```

```r
# Applying the transformation to the dataset
Glass_transformed_ss <- predict(preProc, Glass[, -10])

# Viewing the transformed data
head(Glass_transformed_ss)
```

```
##           RI          Na        Mg         Al          Si           K
## 1  0.3862138  0.12608198 0.5551342 -0.3063816 -0.49869486 -0.29720647
## 2 -0.1782866  0.42318047 0.4548937 -0.1218880  0.07316353 -0.01874385
## 3 -0.4739625  0.09851782 0.3951817  0.1254439  0.28831726 -0.10811108
## 4 -0.1932662 -0.20158533 0.5799799 -0.2581474 -0.04397201  0.09305653
## 5 -0.2268925 -0.12303180 0.4726731 -0.2991170  0.40373539  0.05916450
## 6 -0.3036441 -0.29004891 0.2459643  0.1344417  0.15791804  0.08402131
##            Ca         Ba         Fe
## 1 -0.06449657 -0.1561358 -0.2594843
## 2 -0.56756147 -0.2523255 -0.4193433
## 3 -0.54468469 -0.2318677 -0.3853442
## 4 -0.43085072 -0.2929133 -0.4867968
## 5 -0.45422808 -0.2565822 -0.4264175
## 6 -0.23892010 -0.1349600  0.7986264
```

```r
# Plotting distributions after spatialSign transformation
Glass_long_ss <- reshape2::melt(Glass_transformed_ss)
ggplot(Glass_long_ss, aes(value)) +
  geom_histogram(bins = 30, fill = "lightblue", color = "black") +
  facet_wrap(~variable, scales = "free") +
  theme_bw() +
  labs(title = "Distributions After SpatialSign Transformation")
```




The *spatialSign* transformation was applied to all predictors because it normalizes multivariate data, mitigating the effect of outliers and skewed distributions. Given that several predictors, such as *Mg*, *K*, *Ba*, and *Fe*, show significant skewness, this transformation helps ensure that all variables are on a comparable scale, which can improve the performance of classification models by making the data more robust and consistent.

## Exercise 3.2: Soybean Dataset Analysis

### (a) Investigating Frequency Distributions

```
# Load the Soybean dataset
data(Soybean)

# Summary of the dataset structure
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 ...
##  $ precip         : Ord.factor w/ 3 levels "0"<"1"<"2": 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 ...
##  $ 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 ...
```

```r
# Investigate frequency distributions for categorical variables
# Using lapply to check the frequency of levels for each categorical predictor
freq_distributions <- lapply(Soybean, table)
freq_distributions
```

```
## $Class
##
##               2-4-d-injury         alternarialeaf-spot
##                         16                          91
##                anthracnose            bacterial-blight
##                         44                          20
##          bacterial-pustule                 brown-spot
##                         20                          92
##             brown-stem-rot               charcoal-rot
##                         44                          20
##              cyst-nematode diaporthe-pod-&-stem-blight
##                         14                          15
##       diaporthe-stem-canker               downy-mildew
##                         20                          20
##         frog-eye-leaf-spot            herbicide-injury
##                         91                           8
##      phyllosticta-leaf-spot            phytophthora-rot
##                         20                          88
##             powdery-mildew          purple-seed-stain
##                         20                          20
##        rhizoctonia-root-rot
##                         20
##
## $date
```

```
##
##   0   1   2   3   4   5   6
##  26  75  93 118 131 149  90
##
## $plant.stand
##
##   0   1
## 354 293
##
## $precip
##
##   0   1   2
##  74 112 459
##
## $temp
##
##   0   1   2
##  80 374 199
##
## $hail
##
##   0   1
## 435 127
##
## $crop.hist
##
##   0   1   2   3
##  65 165 219 218
##
## $area.dam
##
##   0   1   2   3
## 123 227 145 187
##
## $sever
##
##   0   1   2
## 195 322  45
##
## $seed.tmt
##
##   0   1   2
## 305 222  35
##
## $germ
##
##   0   1   2
## 165 213 193
##
## $plant.growth
##
##   0   1
## 441 226
##
```

```
## $leaves
##
##   0   1
##  77 606
##
## $leaf.halo
##
##   0   1   2
## 221  36 342
##
## $leaf.marg
##
##   0   1   2
## 357  21 221
##
## $leaf.size
##
##   0   1   2
##  51 327 221
##
## $leaf.shread
##
##   0   1
## 487  96
##
## $leaf.malf
##
##   0   1
## 554  45
##
## $leaf.mild
##
##   0   1   2
## 535  20  20
##
## $stem
##
##   0   1
## 296 371
##
## $lodging
##
##   0   1
## 520  42
##
## $stem.cankers
##
##   0   1   2   3
## 379  39  36 191
##
## $canker.lesion
##
##   0   1   2   3
## 320  83 177  65
```

```
##
## $fruiting.bodies
##
##   0   1
## 473 104
##
## $ext.decay
##
##   0   1   2
## 497 135  13
##
## $mycelium
##
##   0   1
## 639   6
##
## $int.discolor
##
##   0   1   2
## 581  44  20
##
## $sclerotia
##
##   0   1
## 625  20
##
## $fruit.pods
##
##   0   1   2   3
## 407 130  14  48
##
## $fruit.spots
##
##   0   1   2   4
## 345  75  57 100
##
## $seed
##
##   0   1
## 476 115
##
## $mold.growth
##
##   0   1
## 524  67
##
## $seed.discolor
##
##   0   1
## 513  64
##
## $seed.size
##
##   0   1
```

```
## 532  59
##
## $shriveling
##
##   0   1
## 539  38
##
## $roots
##
##   0   1   2
## 551  86  15
```

```
# Check for degenerate distributions (those with most values in a single category)
degenerate_vars <- sapply(Soybean, function(x) max(table(x)) / length(x))
degenerate_vars
```

```
##           Class            date     plant.stand          precip            temp
##       0.1346999       0.2181552       0.5183016       0.6720351       0.5475842
##            hail       crop.hist        area.dam           sever        seed.tmt
##       0.6368960       0.3206442       0.3323572       0.4714495       0.4465593
##            germ    plant.growth          leaves       leaf.halo       leaf.marg
##       0.3118594       0.6456808       0.8872621       0.5007321       0.5226940
##       leaf.size     leaf.shread       leaf.malf       leaf.mild            stem
##       0.4787701       0.7130307       0.8111274       0.7833089       0.5431918
##         lodging    stem.cankers   canker.lesion fruiting.bodies       ext.decay
##       0.7613470       0.5549048       0.4685212       0.6925329       0.7276720
##        mycelium    int.discolor       sclerotia      fruit.pods     fruit.spots
##       0.9355783       0.8506589       0.9150805       0.5959004       0.5051245
##            seed     mold.growth   seed.discolor       seed.size      shriveling
##       0.6969253       0.7672035       0.7510981       0.7789165       0.7891654
##           roots
##       0.8067350
```

**Breakdown of Variables and Frequency Distributions:**

- **$shriveling:**
  - 0: 539 occurrences
  - 1: 38 occurrences
  - Proportion of dominant category (0): $\frac{539}{683} \approx 0.79$
  - Not degenerate, but highly skewed towards 0.
- **$roots:**
  - 0: 551 occurrences
  - 1: 86 occurrences
  - 2: 15 occurrences
  - Proportion of dominant category (0): $\frac{551}{683} \approx 0.81$
  - Not degenerate, but skewed.
- **$seed:**
  - 0: 476 occurrences
  - 1: 115 occurrences
  - Proportion of dominant category (0): $\frac{476}{683} \approx 0.70$
  - Not degenerate, but skewed.
- **$mold.growth:**
  - 0: 524 occurrences
  - 1: 67 occurrences
  - Proportion of dominant category (0): $\frac{524}{683} \approx 0.77$

  - Not degenerate, but skewed.
- **$mycelium:**
  - 0: 639 occurrences
  - 1: 6 occurrences
  - Proportion of dominant category (0): $\frac{639}{683} \approx 0.94$
  - **Degenerate variable** (since the proportion exceeds 90%).
- **$int.discolor:**
  - 0: 581 occurrences
  - 1: 44 occurrences
  - 2: 20 occurrences
  - Proportion of dominant category (0): $\frac{581}{683} \approx 0.85$
  - Not degenerate, but highly skewed.
- **$sclerotia:**
  - 0: 625 occurrences
  - 1: 20 occurrences
  - Proportion of dominant category (0): $\frac{625}{683} \approx 0.91$
  - **Degenerate variable** (exceeds 90%).

**Summary of Degenerate Variables:**

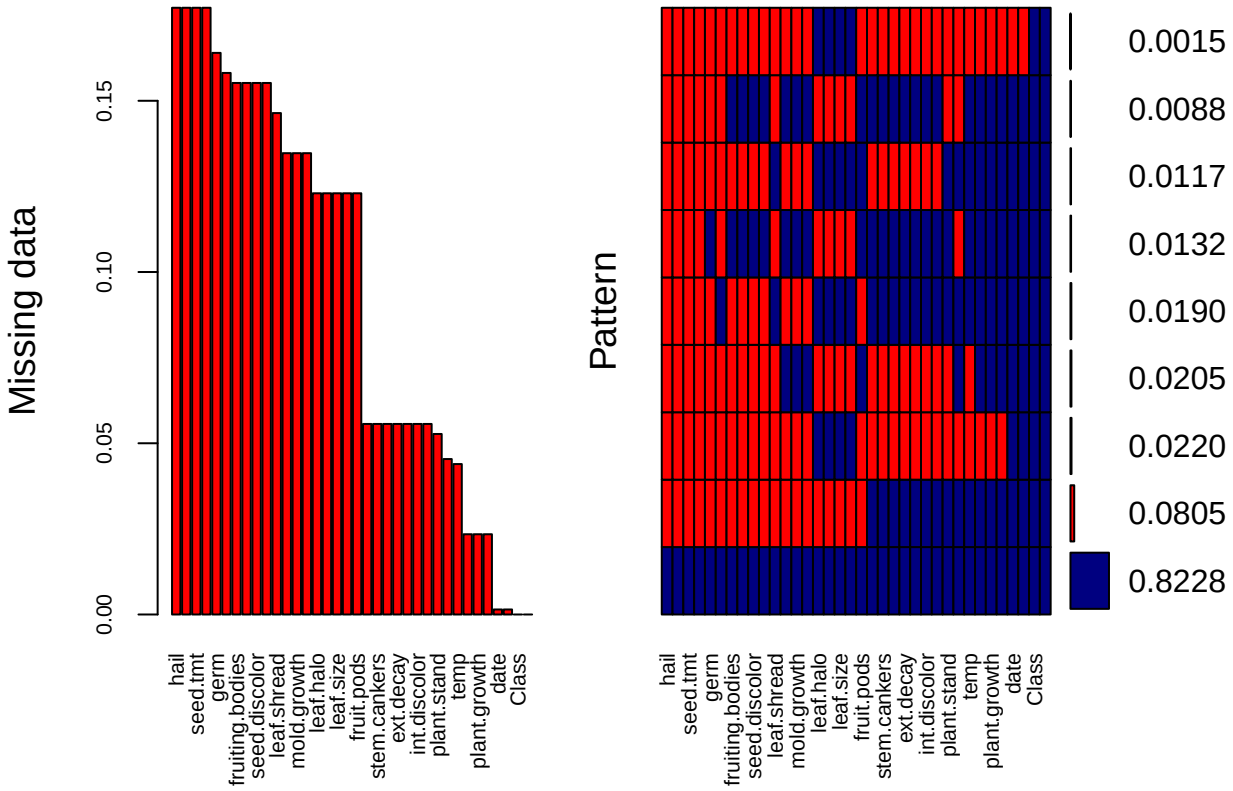
- **$mycelium:** 93.6% of values are in category 0.
- **$sclerotia:** 91.5% of values are in category 0.
- These two variables are degenerate, as more than 90% of their values fall into a single category.
- Other variables like $shriveling, $roots, and $mold.growth show skewness but are not considered degenerate by the 90% threshold.

### (b) Investigating Missing Data

```
# Investigating missing data
missing_data_summary <- colSums(is.na(Soybean))
missing_data_summary
```

```
##           Class            date     plant.stand          precip            temp
##               0               1              36              38              30
##            hail       crop.hist        area.dam           sever        seed.tmt
##             121              16               1             121             121
##            germ    plant.growth          leaves       leaf.halo       leaf.marg
##             112              16               0              84              84
##       leaf.size     leaf.shread       leaf.malf       leaf.mild            stem
##              84             100              84             108              16
##         lodging    stem.cankers   canker.lesion fruiting.bodies       ext.decay
##             121              38              38             106              38
##        mycelium    int.discolor       sclerotia      fruit.pods     fruit.spots
##              38              38              38              84             106
##            seed     mold.growth   seed.discolor       seed.size      shriveling
##              92              92             106              92             106
##           roots
##              31
```

```
# Visualizing missing data pattern
library(VIM)
aggr_plot <- aggr(Soybean, col = c('navyblue', 'red'), numbers = TRUE, sortVars = TRUE, labels = names(
```




```
##
##  Variables sorted by number of missings:
##         Variable       Count
##             hail 0.177159590
##            sever 0.177159590
##         seed.tmt 0.177159590
##          lodging 0.177159590
##             germ 0.163982430
##        leaf.mild 0.158125915
##  fruiting.bodies 0.155197657
##      fruit.spots 0.155197657
##    seed.discolor 0.155197657
##       shriveling 0.155197657
##      leaf.shread 0.146412884
##             seed 0.134699854
##      mold.growth 0.134699854
##        seed.size 0.134699854
##        leaf.halo 0.122986823
##        leaf.marg 0.122986823
##        leaf.size 0.122986823
##        leaf.malf 0.122986823
##       fruit.pods 0.122986823
##           precip 0.055636896
##     stem.cankers 0.055636896
##    canker.lesion 0.055636896
##        ext.decay 0.055636896
##         mycelium 0.055636896
##     int.discolor 0.055636896
##        sclerotia 0.055636896
```

```
##      plant.stand 0.052708638
##            roots 0.045387994
##             temp 0.043923865
##        crop.hist 0.023426061
##     plant.growth 0.023426061
##             stem 0.023426061
##             date 0.001464129
##         area.dam 0.001464129
##            Class 0.000000000
##           leaves 0.000000000
```

```
# Checking if missing data is related to class
missing_by_class <- Soybean %>%
  group_by(Class) %>%
  summarise_all(~sum(is.na(.)))
missing_by_class
```

```
## # A tibble: 19 x 36
##    Class   date plant.stand precip  temp  hail crop.hist area.dam sever seed.tmt
##    <fct>  <int>       <int>  <int> <int> <int>     <int>    <int> <int>    <int>
##  1 2-4-d~     1          16     16    16    16        16        1    16       16
##  2 alter~     0           0      0     0     0         0        0     0        0
##  3 anthr~     0           0      0     0     0         0        0     0        0
##  4 bacte~     0           0      0     0     0         0        0     0        0
##  5 bacte~     0           0      0     0     0         0        0     0        0
##  6 brown~     0           0      0     0     0         0        0     0        0
##  7 brown~     0           0      0     0     0         0        0     0        0
##  8 charc~     0           0      0     0     0         0        0     0        0
##  9 cyst-~     0          14     14    14    14         0        0    14       14
## 10 diapo~     0           6      0     0    15         0        0    15       15
## 11 diapo~     0           0      0     0     0         0        0     0        0
## 12 downy~     0           0      0     0     0         0        0     0        0
## 13 frog-~     0           0      0     0     0         0        0     0        0
## 14 herbi~     0           0      8     0     8         0        0     8        8
## 15 phyll~     0           0      0     0     0         0        0     0        0
## 16 phyto~     0           0      0     0    68         0        0    68       68
## 17 powde~     0           0      0     0     0         0        0     0        0
## 18 purpl~     0           0      0     0     0         0        0     0        0
## 19 rhizo~     0           0      0     0     0         0        0     0        0
## # i 26 more variables: germ <int>, plant.growth <int>, leaves <int>,
## #   leaf.halo <int>, leaf.marg <int>, leaf.size <int>, leaf.shread <int>,
## #   leaf.malf <int>, leaf.mild <int>, stem <int>, lodging <int>,
## #   stem.cankers <int>, canker.lesion <int>, fruiting.bodies <int>,
## #   ext.decay <int>, mycelium <int>, int.discolor <int>, sclerotia <int>,
## #   fruit.pods <int>, fruit.spots <int>, seed <int>, mold.growth <int>,
## #   seed.discolor <int>, seed.size <int>, shriveling <int>, roots <int>
```

**Missing Data Observations:**

- The variables with the highest proportions of missing values are *hail*, *seed.tmt*, *sever*, and *germ*, each with approximately 17.7% missing data.
- Variables such as *lodging*, *leaf.mild*, and *fruiting.bodies* also have over 15% of their data missing.
- The distribution of missing data across various variables shows some patterns, as visualized in the plots, where some variables exhibit clusters of missingness in the dataset.

**Pattern of Missing Data:**

- The heatmap of the missing data pattern indicates that certain combinations of variables have missing data together, which might point to systematic missingness related to specific conditions.

### (c) Handling Missing Data

```
# Loading necessary libraries
library(caret)
library(dplyr)

# Separating numeric and categorical variables
numericVars <- Soybean[, sapply(Soybean, is.numeric)]
categoricalVars <- Soybean[, sapply(Soybean, is.factor)]

# Imputation: numeric variables using median impute
preProcNumeric <- preProcess(numericVars, method = "medianImpute")
numeric_imputed <- predict(preProcNumeric, numericVars)

# Mode imputation for categorical variables
mode_impute <- function(x) {
  x[is.na(x)] <- names(which.max(table(x)))
  return(x)
}
categorical_imputed <- categoricalVars %>% mutate_all(mode_impute)

# Combining the imputed datasets
Soybean_imputed <- cbind(numeric_imputed, categorical_imputed)

# Checking the dataset after imputation
print("Number of missing data in Soybean after imputation:"); sum(is.na(Soybean_imputed))
```

```
## [1] "Number of missing data in Soybean after imputation:"
```

```
## [1] 0
```

- For handling missing data in the Soybean dataset, we applied imputation techniques to both numerical and categorical variables.
- **Numerical Variables**: We used median imputation to replace missing values, as it is less sensitive to outliers and ensures that the central tendency of the data is preserved.
- **Categorical Variables**: Mode imputation was applied, replacing missing values with the most frequent category, ensuring that the distribution of the categorical variables remains representative.
- After imputation, no missing values remain in the dataset, as confirmed by a final check which returned zero missing entries. This ensures the dataset is complete for subsequent analysis.

---