**KJ 3.2:** The soybean data can 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.

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

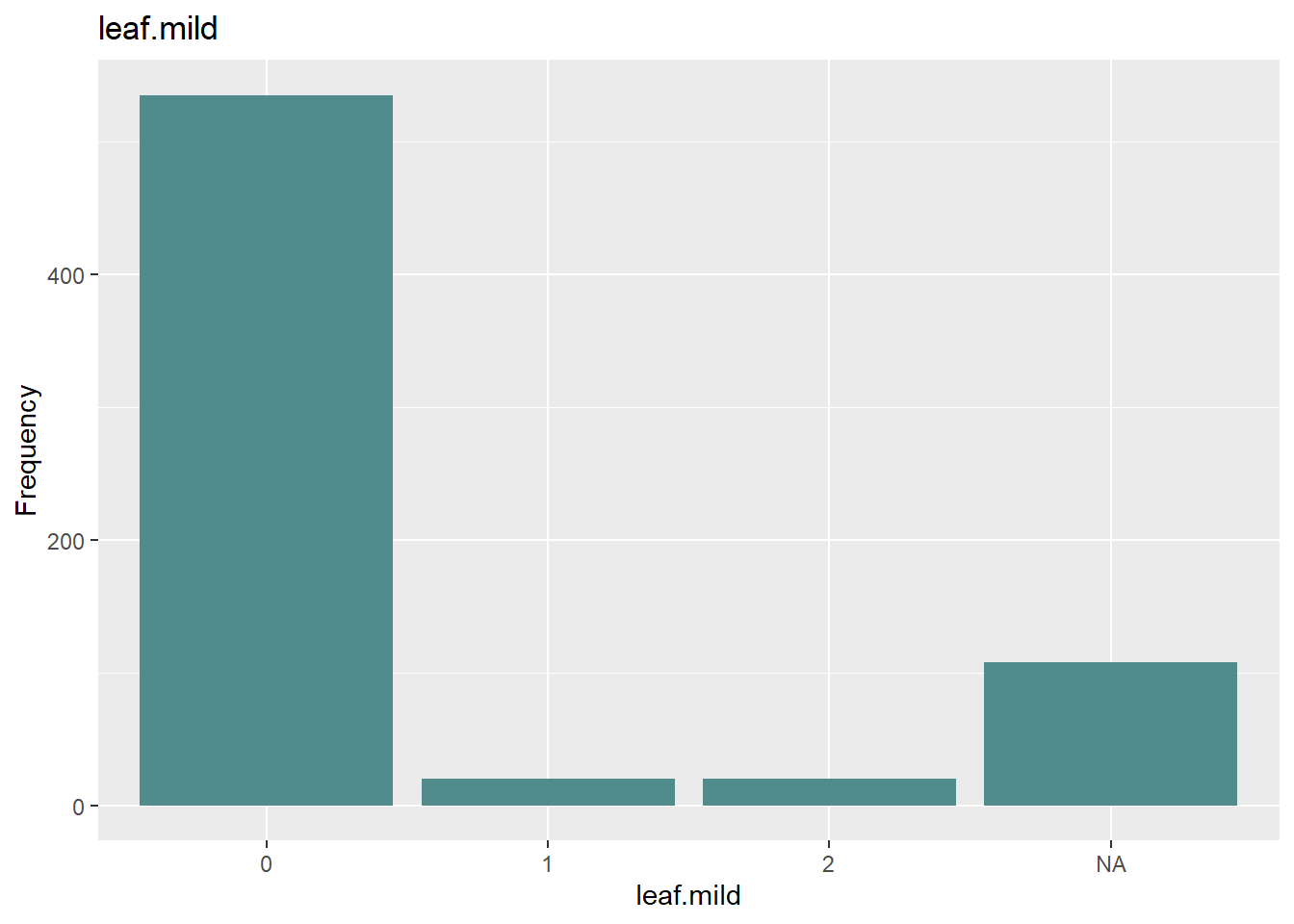
There were no “zero variation” variables in this dataset, as all predictors had more than one unique value. However, three variables exhibited “near-zero variation”: their ratio of most-common to second-most-common variables exceeded 20:1 (Kuhn, 44). Those three variables were:

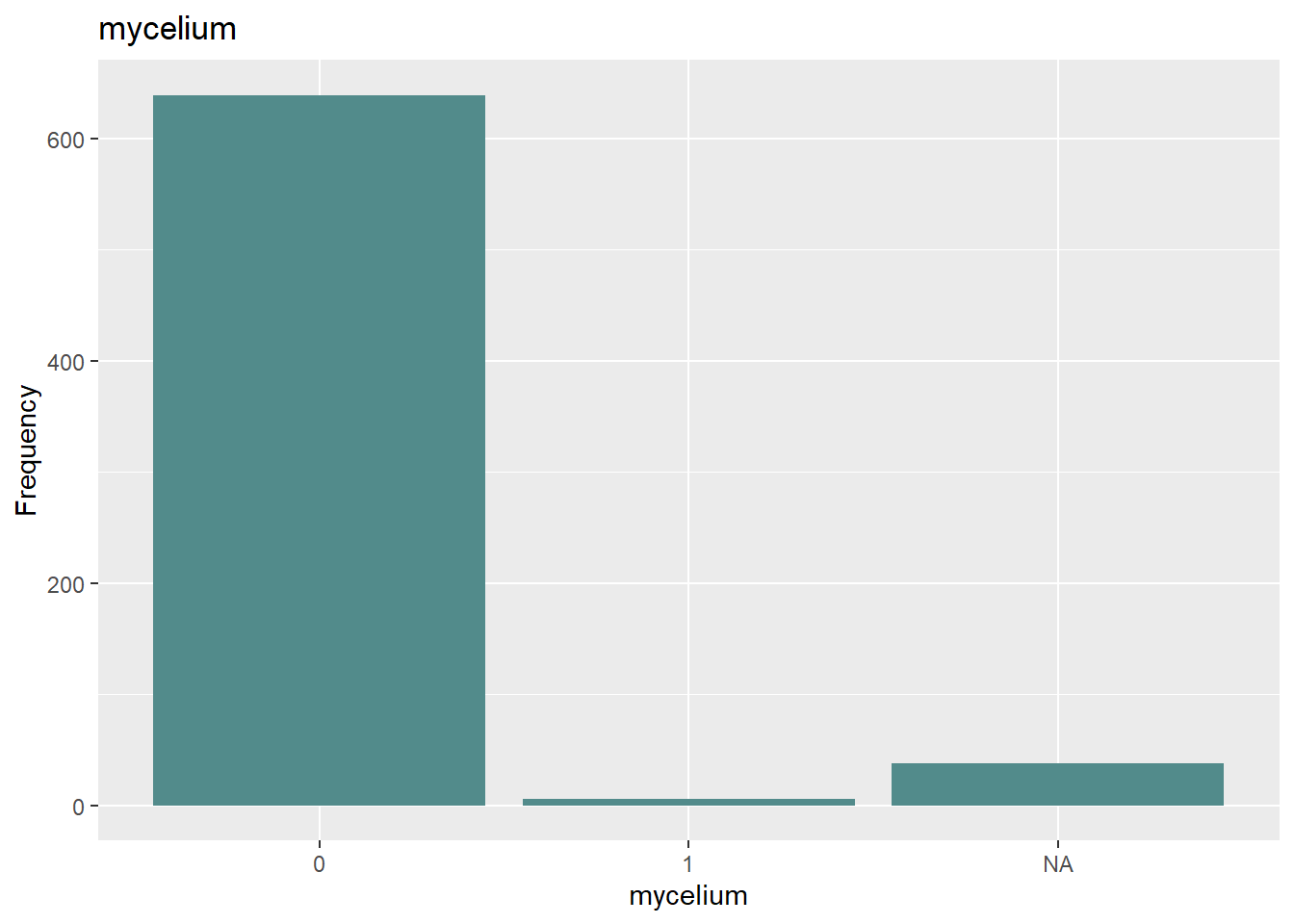
[1] "leaf.mild" "mycelium" "sclerotia"

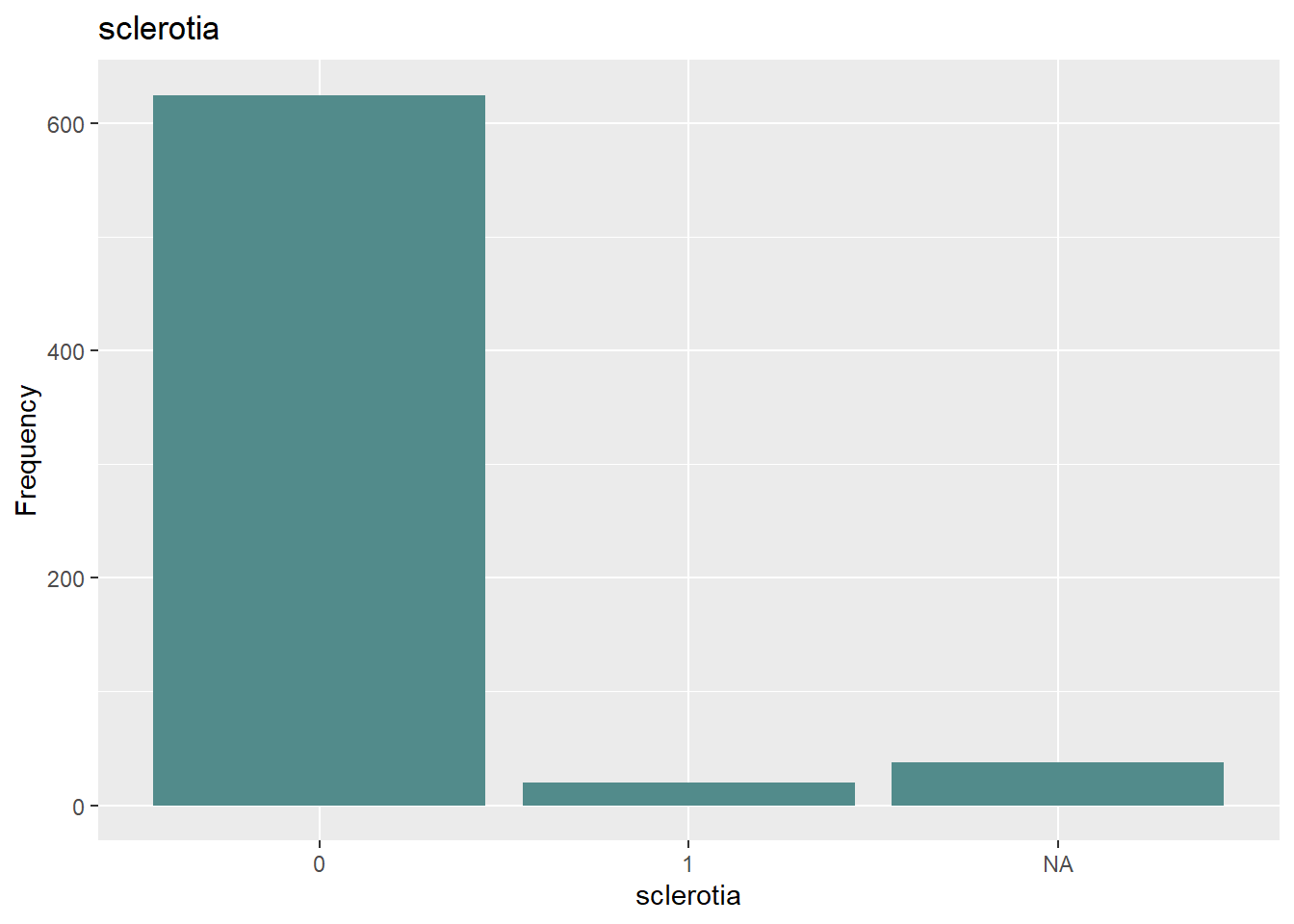
The table and bar charts below display the frequency of the most-common vs. the second-most-common values for each of these three degenerate variables.

# A tibble: 3 × 6

var most\_freq most\_freq\_count sec\_most\_freq sec\_most\_freq\_count ratio  
 <chr> <chr> <dbl> <chr> <dbl> <dbl>  
1 leaf.mild 0 535 1 20 26.8  
2 mycelium 0 639 1 6 106.0   
3 sclerotia 0 625 1 20 31.2

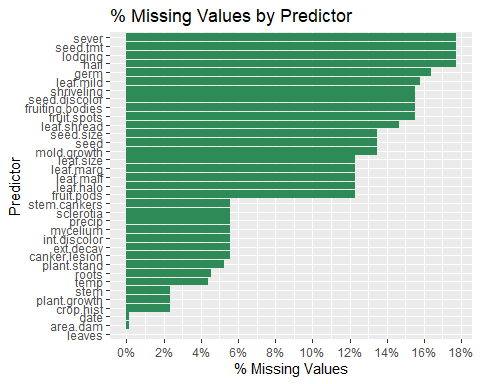






**B. Roughly 18% of the data are missing. Are there particular predictors that are more likely to be missing? Is the pattern of missing data related to the classes?**

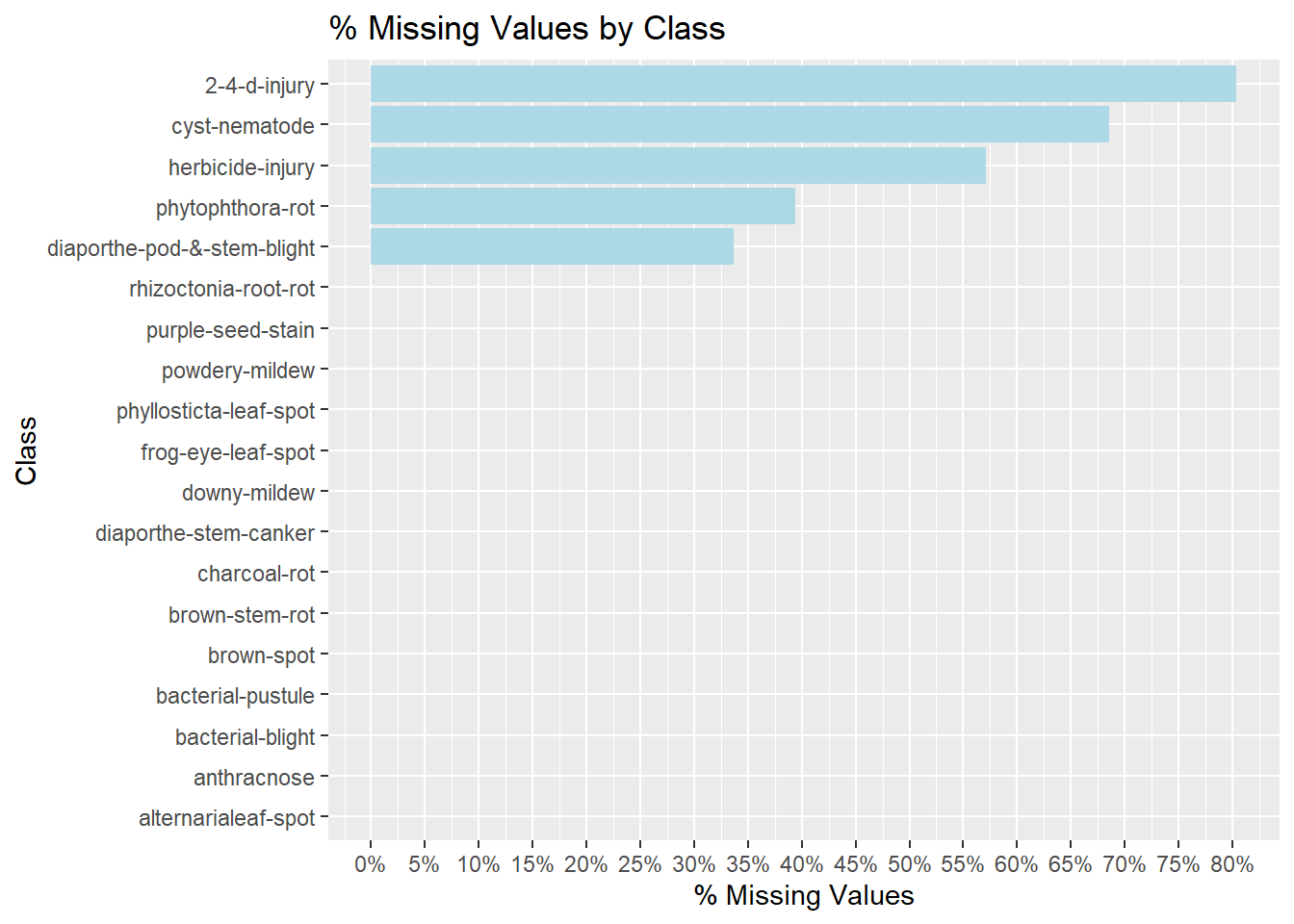
Predictors: Yes, some predictors are much more likely than others to have missing values. The bar chart below shows the rate of missing data for all 35 predictor variables. Note that only one variable (leaves) has no missing values at all, while the rest range widely from <1% (area.dam, date) to 17.73% (sever, seed.tmt, lodging, hail).



Classes: There is also a strong relationship between the outcome variable (class) and missing data. The summary grid and bar chart below illustrate the proportion of NA predictor values to total possible predictor values for each Class (number of rows \* 35 total predictor variables).

Only five Classes had any missing values, but these were all quite significant, ranging from 33.7% to 80.4% of the total possible values.

# A tibble: 5 × 5  
 Class total\_rows total\_values total\_na pct\_na  
 <fct> <int> <dbl> <dbl> <dbl>  
1 2-4-d-injury 16 560 450 0.804  
2 cyst-nematode 14 490 336 0.686  
3 herbicide-injury 8 280 160 0.571  
4 phytophthora-rot 88 3080 1214 0.394  
5 diaporthe-pod-&-stem-blight 15 525 177 0.337



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

To develop an effective strategy to handle missing data, one should carefully evaluate the data and engage with the subject matter experts to understand the context. Since there is a strong relationship between the NA values and the Classes, at least some of the missing data is likely to be meaningful and should not be eliminated without due consideration.

For example, it may be that a value is missing for a given variable (e.g. “fruiting bodies”) because the plant never matured: in cases like this, the missing data can considered “informative missing” and should be included.

Alternately, a missing value may indicate a value beyond a measurable threshold (“censored missing”). In this case, as above, the data is still meaningful and should receive a value equal to the threshold or a random number between zero and the threshold value.

Or a value might simply be missing due to data quality, collection, or other reasons, and such a variable might be imputed using K-nearest neighbor or correlating it to another variable, if a strong correlation exists.

Of course, if the predictor with significant missing data is highly correlated to another variable, it might also be eliminated as redundant to reduce complexity, if no additional value would be gained by keeping it.

**Works Cited:**

Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed June 2024.

Kuhn, Max, et al. Applied Predictive Modeling. New York, Springer, 2016.

**Code for reference:**

# Load libraries

library(tidyverse)

library(dplyr)

library(caret)

library(ggplot2)

library(forcats)

library(fpp2)

library(mlbench)

library(RColorBrewer)

library(scales)

# Load Soybean dataset

data(Soybean)

# Create df excluding outcome variable "Class"

soy\_data <- select(Soybean,-Class)

#--------------------------

# Section A:

#--------------------------

#----------------------------------------

# Zero-Var:

#----------------------------------------

# Identify Zero-Var: Count unique values for each column and extract those where count = 1

unique\_values <- soy\_data %>%

summarise\_all(n\_distinct)

zero\_var <- names(unique\_values)[unique\_values == 1]

zero\_var

#----------------------------------------

# Near-Zero-Var:

#----------------------------------------

# Identify Near-Zero-Var: Use function in caret pkg

near\_zero <- soy\_data %>%

nearZeroVar()

near\_zero\_vars <- (names(soy\_data)[near\_zero])

near\_zero\_vars

# Calculate ratios for the above: Create df to hold the ratios

df\_soy\_freq <- tibble(

var = character(),

most\_freq = character(),

most\_freq\_count = numeric(),

sec\_most\_freq = character(),

sec\_most\_freq\_count = numeric(),

ratio = numeric()

)

# Iterate through the near-zero-var variables above:

for (col\_name in (names(soy\_data)[near\_zero])) {

soy\_var <- soy\_data[[col\_name]]

soy\_var\_freq <- table(soy\_var) %>%

sort(decreasing = TRUE)

freq\_1 <- soy\_var\_freq[1]

freq\_2 <- soy\_var\_freq[2]

freq\_ratio <- freq\_1/freq\_2

df\_soy\_freq <- bind\_rows(df\_soy\_freq,tibble(

var = col\_name,

most\_freq = names(soy\_var\_freq[1]),

most\_freq\_count = freq\_1,

sec\_most\_freq = names(soy\_var\_freq[2]),

sec\_most\_freq\_count = freq\_2,

ratio = freq\_ratio

)

)

}

#df\_soy\_freq$ratio <- as.numeric(df\_soy\_freq$ratio)

df\_soy\_freq

# Create bar chart for each degenerate variable

for (i in near\_zero\_vars) {

soy\_plot <- ggplot(soy\_data, aes\_string(x=i)) +

geom\_bar(fill = "darkslategray4") +

ggtitle(i) +

xlab(i) +

ylab('Frequency')

print(soy\_plot)

}

#--------------------------

# Section B:

#--------------------------

#--------------------------

# Missing Values by Predictor:

#--------------------------

missing\_pred <- soy\_data %>%

summarise\_all(~sum(is.na(.))/n()) %>%

pivot\_longer(everything(),names\_to = "predictor", values\_to = "pct\_na") %>%

arrange(desc(pct\_na))

tail(missing\_pred)

missing\_plot <- missing\_pred %>%

ggplot(aes(x = reorder(predictor, pct\_na), y = pct\_na)) +

geom\_col(fill = "seagreen") +

coord\_flip() +

scale\_y\_continuous(

labels = scales::percent,

breaks = breaks\_width(.02,0)) +

labs(

title = "% Missing Values by Predictor",

x = "Predictor",

y = "% Missing Values")

missing\_plot

#--------------------------

# Missing Values by Class:

#--------------------------

# add column to count of missing values across all predictor variables

missing\_outcomes <- Soybean %>%

mutate(

count\_na = rowSums(is.na(Soybean[,-1])),

count\_total = 35) %>%

group\_by(Class) %>%

summarise(

total\_rows = n(),

total\_values = sum(count\_total),

total\_na= sum(count\_na),

pct\_na = total\_na/total\_values) %>%

arrange(desc(pct\_na))

head(missing\_outcomes, 5)

# plot distribution of n/a values by class

missing\_plot2 <- missing\_outcomes %>%

ggplot(aes(x = reorder(Class, pct\_na), y = pct\_na)) +

geom\_col(fill = "lightblue") +

coord\_flip() +

scale\_y\_continuous(

labels = scales::percent,

breaks = breaks\_width(.05,0)) +

labs(

title = "% Missing Values by Class",

x = "Class",

y = "% Missing Values")

missing\_plot2