Data 624 Homework 4 Chapter 3.1

Enid Roman

2024-09-28

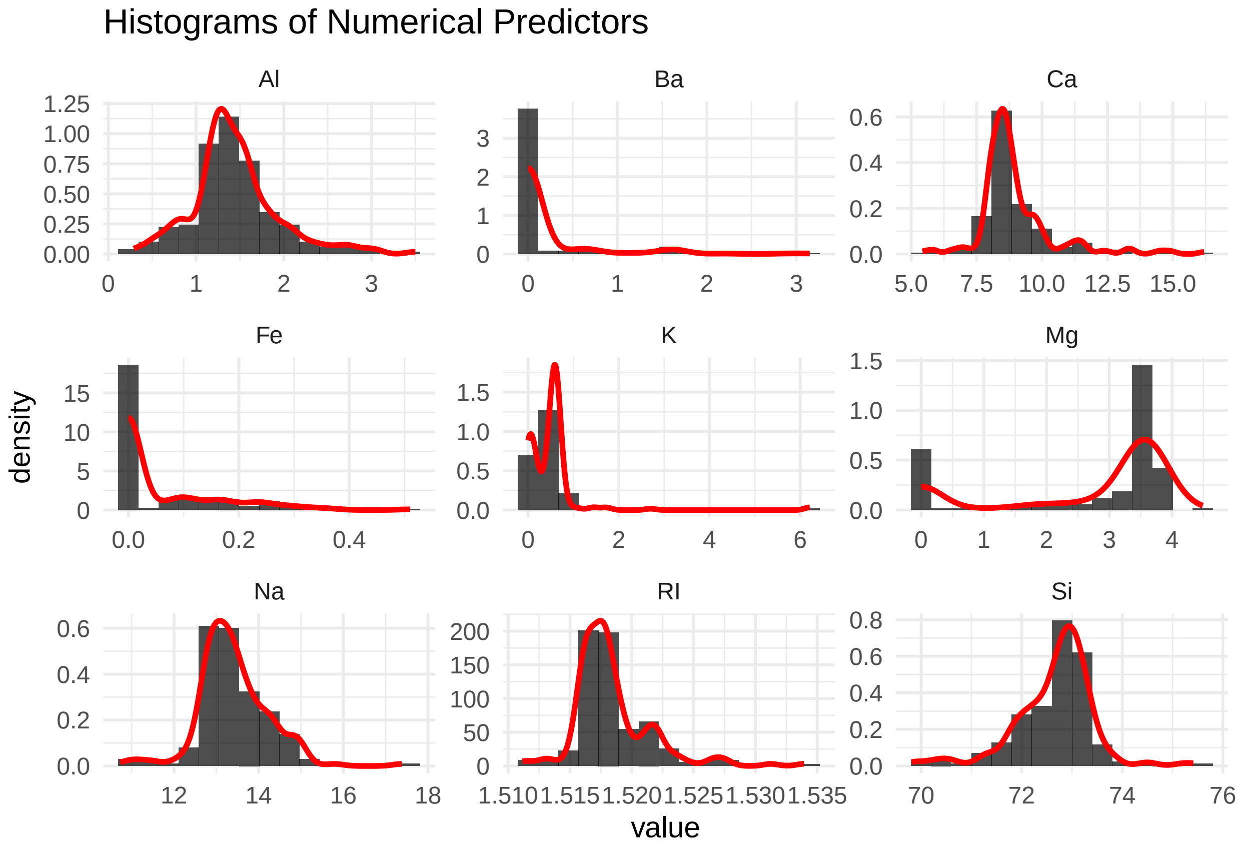
3.1. The UC Irvine Machine Learning Repository6 contains a data set related to glass identification. The data consist of 214 glass samples labeled as one of seven class categories. There are nine predictors, including the refractive index and percentages of eight elements: Na, Mg, Al, Si, K, Ca, Ba, and Fe. The data can be accessed via:

```r
library(ggplot2)
library(gridExtra)
library(dplyr)
library(tidyverse)
library(corrplot)
library(tidyr)


#install.packages('mlbench')
library(mlbench)
data(Glass)
str(Glass)
```

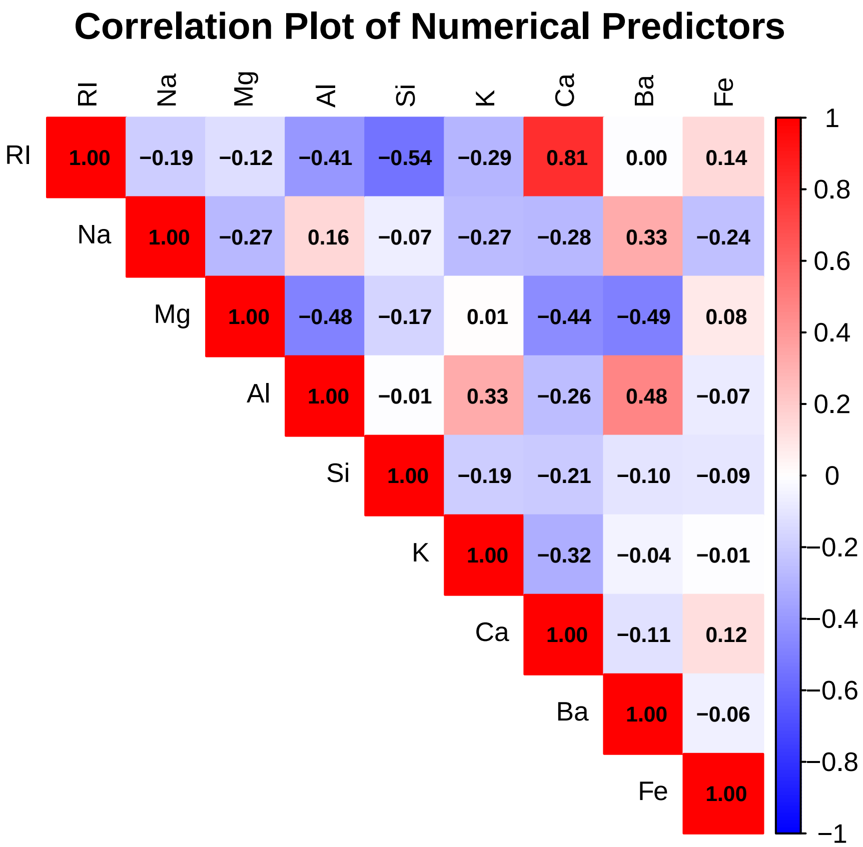
```
## ’data.frame':    214 obs. of  10 variables:
##  $ RI  : num  1.52 1.52 1.52 1.52 1.52 ...
##  $ Na  : num  13.6 13.9 13.5 13.2 13.3 ...
##  $ Mg  : num  4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...
##  $ Al  : num  1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...
##  $ Si  : num  71.8 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.

```r
# Create individual histograms for each predictor with density line
Glass %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
  geom_histogram(aes(y=..density..), bins = 15, fill="black", alpha=0.7) +
  geom_density(color="red", size=1) +
  facet_wrap(~key, scales = 'free') +
  ggtitle("Histograms of Numerical Predictors") +
  theme_minimal()
```

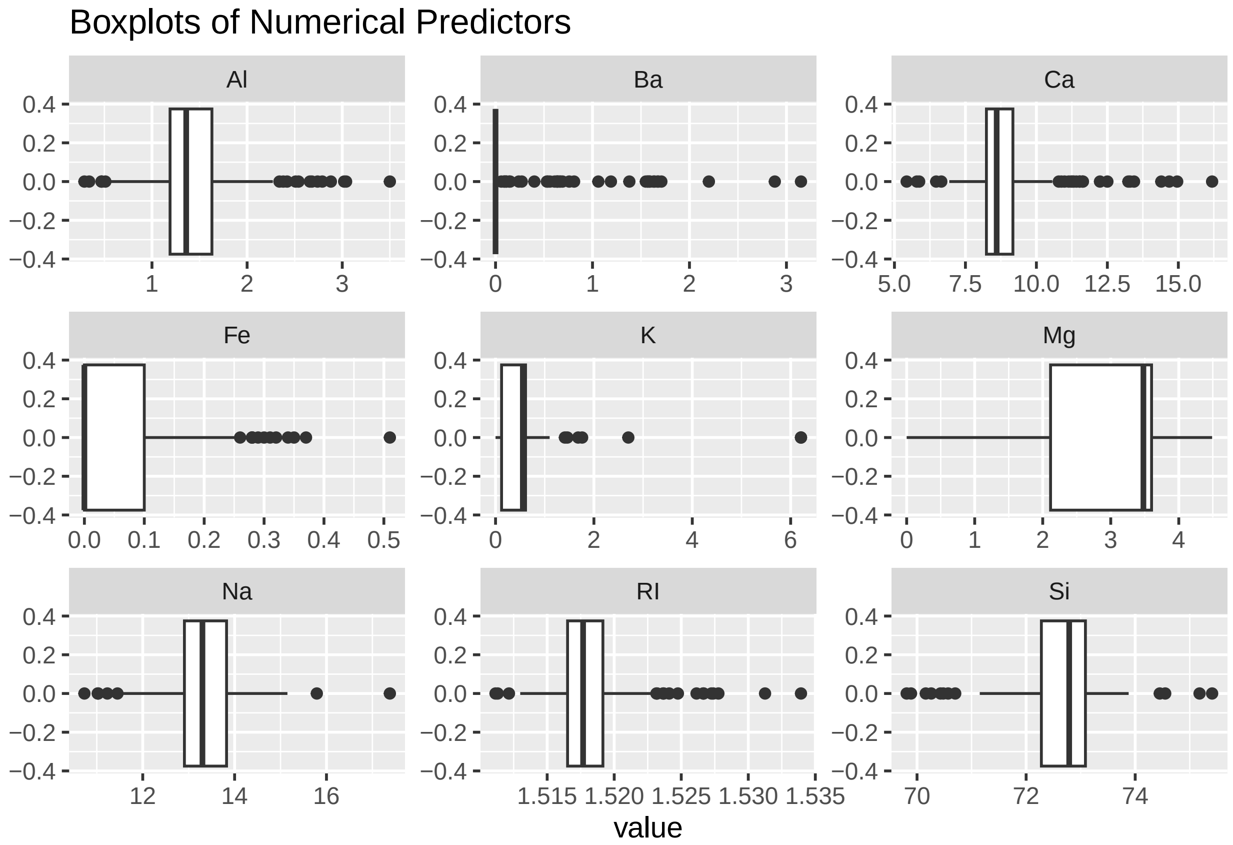


```
# Compute the correlation matrix for numerical variables
cor_matrix <- cor(Glass[, sapply(Glass, is.numeric)])

# Generate the correlation plot
corrplot(cor_matrix, method = "color", type = "upper",
         tl.col = "black", tl.cex = 0.8,
         addCoef.col = "black", number.cex = 0.7,
         col = colorRampPalette(c("blue", "white", "red"))(200),
         title = "Correlation Plot of Numerical Predictors",
         mar = c(0, 0, 2, 0))
```



	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe
RI	1.00	−0.19	−0.12	−0.41	−0.54	−0.29	0.81	0.00	0.14
Na		1.00	−0.27	0.16	−0.07	−0.27	−0.28	0.33	−0.24
Mg			1.00	−0.48	−0.17	0.01	−0.44	−0.49	0.08
Al				1.00	−0.01	0.33	−0.26	0.48	−0.07
Si					1.00	−0.19	−0.21	−0.10	−0.09
K						1.00	−0.32	−0.04	−0.01
Ca							1.00	−0.11	0.12
Ba								1.00	−0.06
Fe									1.00

```r
# Create the boxplot for numerical variables
Glass %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
  geom_boxplot() +
  facet_wrap(~key, scales = 'free') +
  ggtitle("Boxplots of Numerical Predictors")
```



Outliers:

Histogram: Na and Ca is Fairly normally distributed with slight skew. Ba and Fe have many values concentrated near zero, suggesting these elements are absent or in trace amounts in most samples.

Correlation: Ca and RI show a strong positive correlation, meaning as calcium increases, the refractive index tends to increase. Mg and Na show a weak negative correlation, suggesting these elements do not vary together in this dataset.

Boxplot: RI and Ca have clear distinctions between different glass types, with certain types having higher values than others. Mg and Ba show that some types of glass have very low or zero amounts, as evidenced by compact boxplots with little spread.

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

Outliers are present in 8 out of the 9 predictors, including RI, Na, Al, Si, K, Ca, Ba, and Fe. The only variable without visible outliers is Mg.

Skewness:

Refractive Index (RI): Slight right-skewness (most values around a central point, slight extension to the right). Sodium (Na): Right-skewed (values concentrated around 13-15, tail toward higher values). Magnesium (Mg): Left-skewed. Most values are higher, and there is a tail extending toward lower values, indicating that a few samples have lower magnesium content. Aluminum (Al): Right-skewed (values between 1 and 2, with a long tail toward higher values). Silicon (Si): Left-skewed (majority of values are high, with a tail extending to lower values). Potassium (K): Strong right-skewness (values near zero, with a long tail toward higher values). Calcium (Ca): Slight right-skewness (fairly uniform, slight skew toward higher values). Barium (Ba): Strong right-skewness (most values near zero, long tail toward higher values). Iron (Fe): Strong right-skewness (majority of values near zero, with a tail toward higher values).

Left-skewed predictors: Si (Silicon) and Mg (Magnesium) are left-skewed, meaning their distributions have longer tails toward lower values, while most of the data is concentrated on the higher end. Right-skewed predictors: Na, Al, K, Ba, and Fe show right-skewness, with a long tail extending toward higher values. Slight right-skewness: RI and Ca show only slight skewness, with a relatively balanced distribution.

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

Applying transformations to some of the skewed predictors can improve the performance of a classification model. Specifically, transformations can help normalize the data, reduce skewness, and mitigate the impact of outliers, which can in turn lead to better model performance. Here's a summary of relevant transformations for the Glass dataset based on the skewness and presence of outliers.

Right-skewed distributions: Transformations like the log, square root, or Box-Cox transformations are commonly used to reduce right skewness. Left-skewed distributions: A reverse log transformation or square root transformation can be used to normalize left-skewed data. Outliers: Transformations can reduce the impact of outliers by compressing the extreme values.

Log transformations are recommended for right-skewed variables (Na, Al, K, Ba, Fe). Reverse log transformations can help with left-skewed variables like Mg and Si. For slightly skewed variables like Ca and RI, a square root transformation would be beneficial.

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:

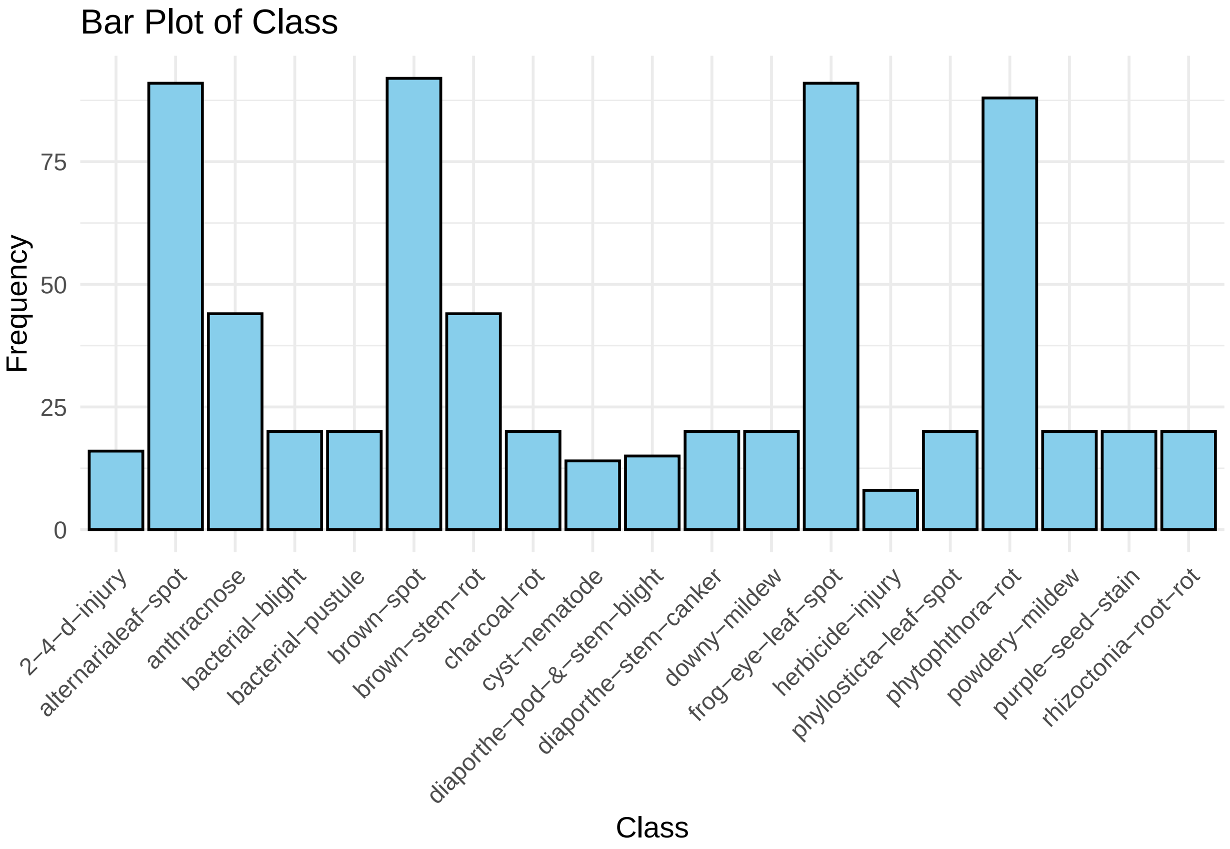
```r
library(mlbench)
data(Soybean)
## See ?Soybean for details
## ?Soybean
```

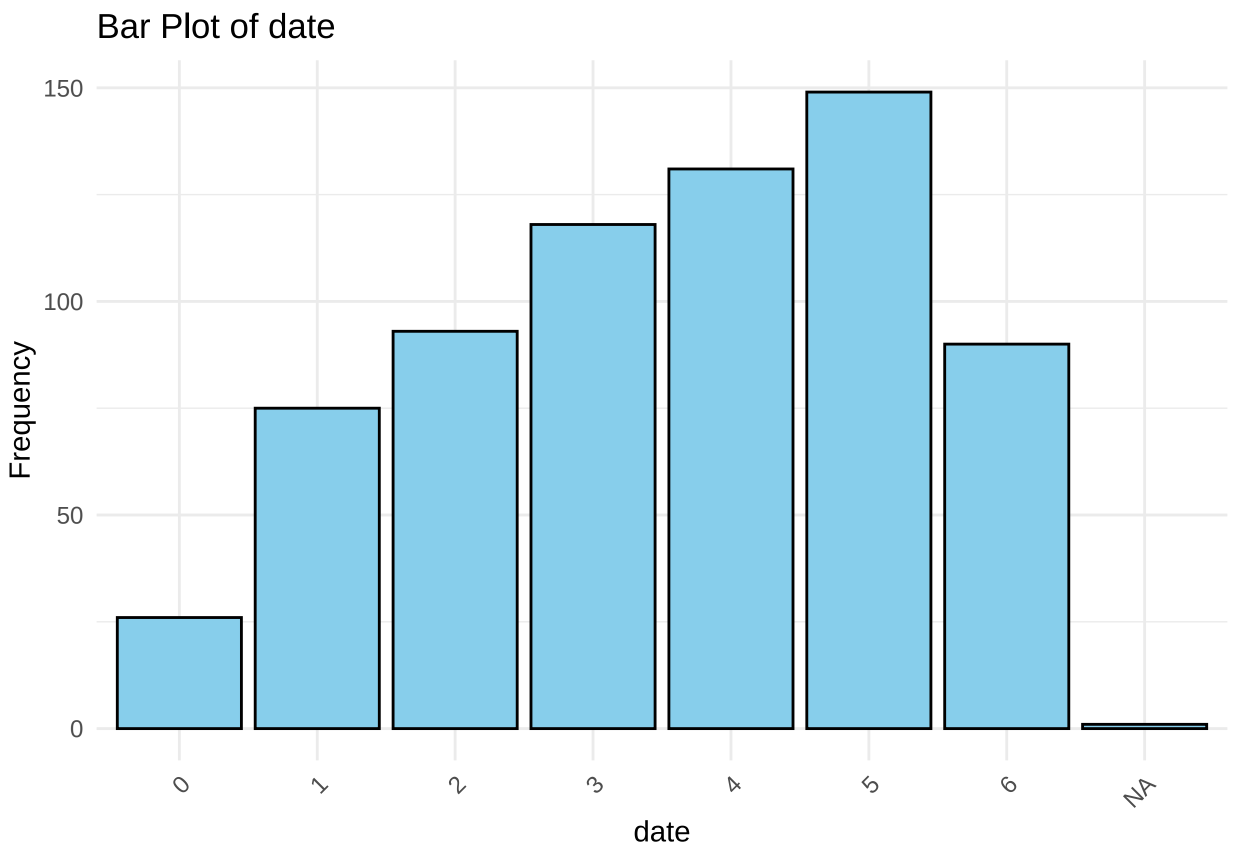
(a) Investigate the frequency distributions for the categorical predictors. Are any of the distributions degenerate in the ways discussed earlier in this chapter?

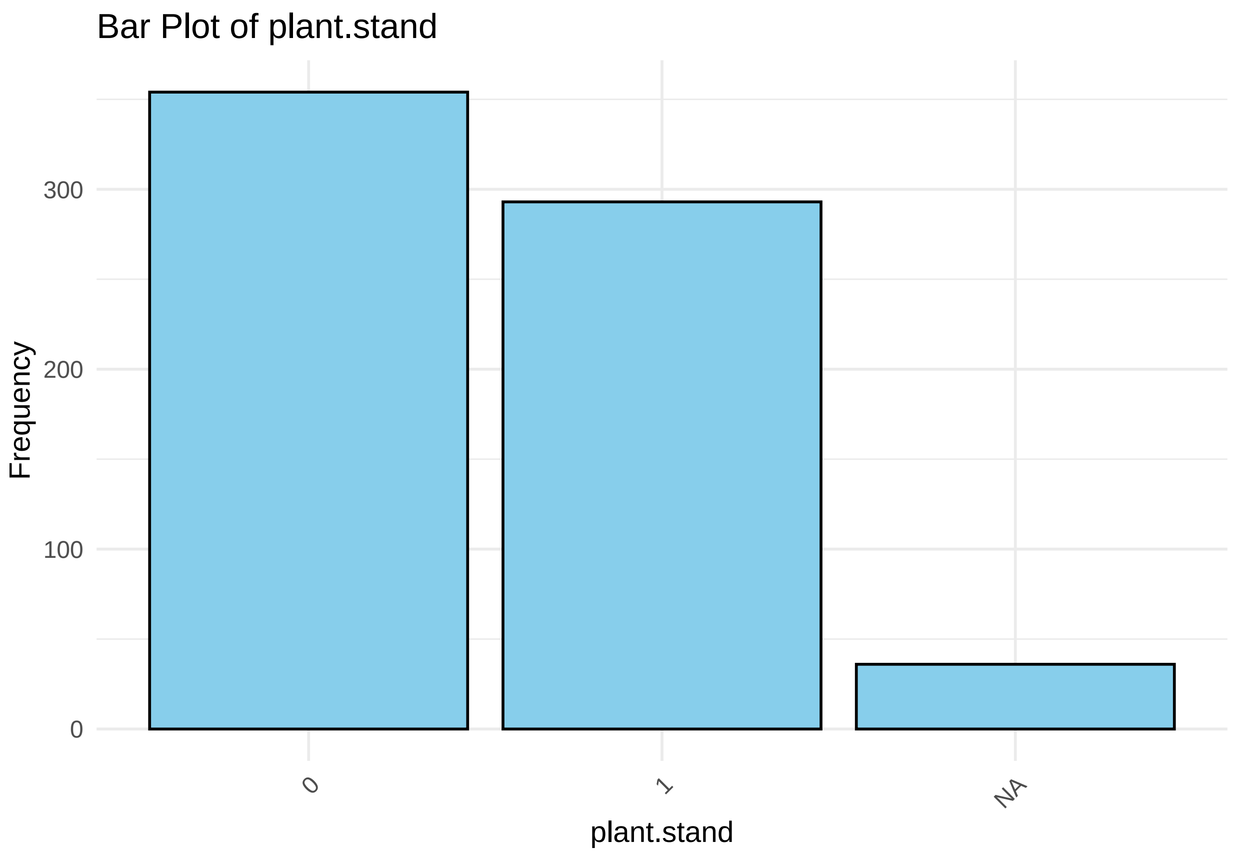
```r
# Check the structure to identify categorical variables
str(Soybean)
```

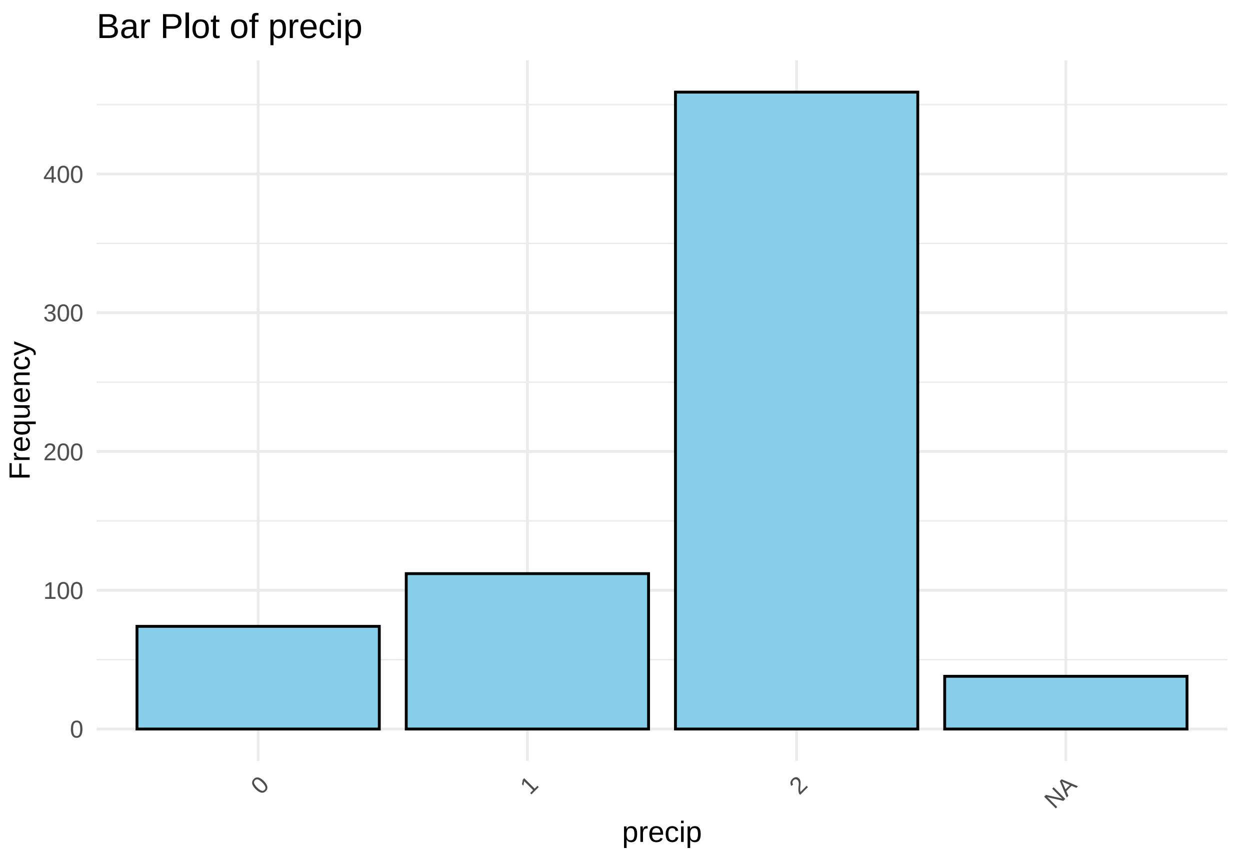
```
## 'data.frame':    683 obs. of  36 variables:
##  $ Class          : Factor w/ 19 levels "2-4-d-injury",..: 11 11 11 11 11 11 11 11 11 11 ...
##  $ date           : Factor w/ 7 levels "0","1","2","3",..: 7 5 4 4 7 6 6 5 7 5 ...
##  $ plant.stand    : Ord.factor w/ 2 levels "0"<"1": 1 1 1 1 1 1 1 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 ...
```

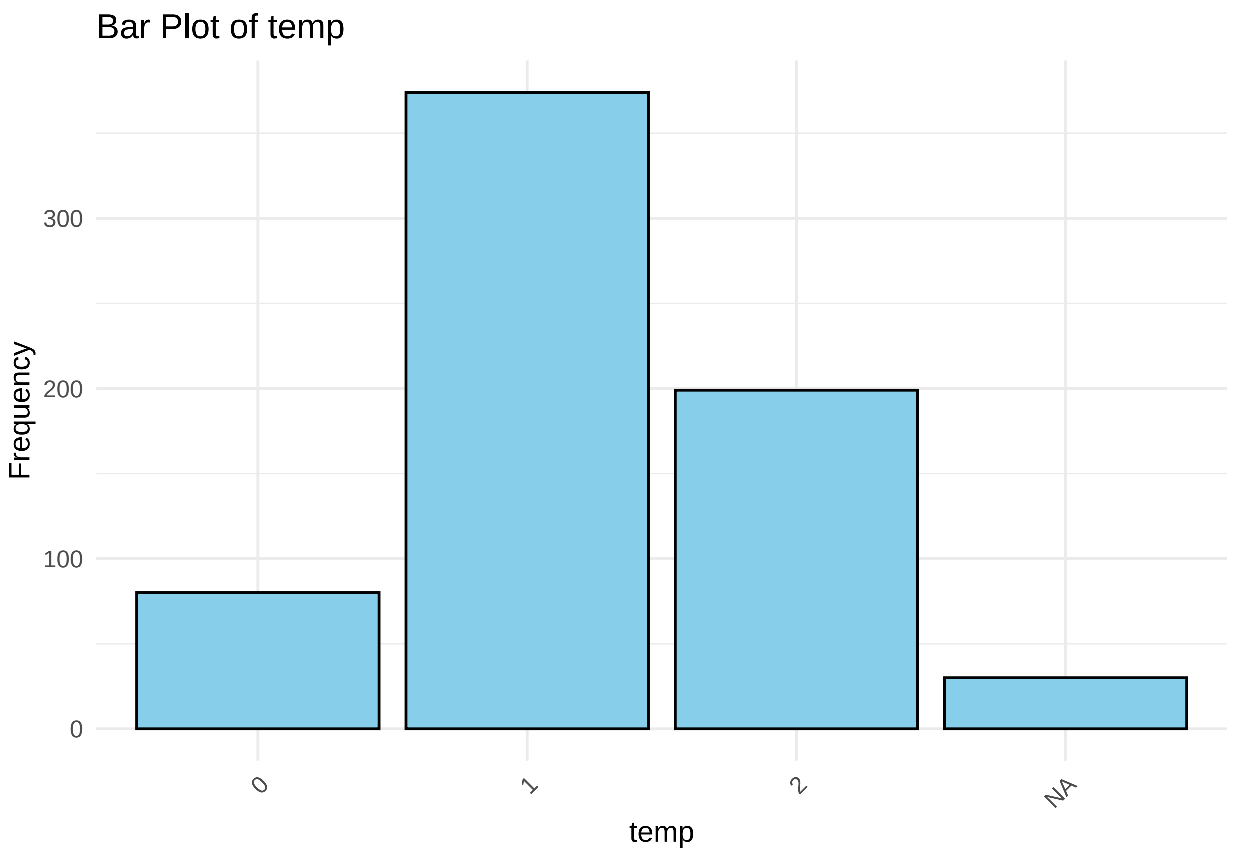
```r
# Identify categorical columns
categorical_columns <- names(Soybean)[sapply(Soybean, is.factor)]

# Create bar plots for each categorical variable
for (col in categorical_columns) {
  # Create the bar plot
  p <- ggplot(Soybean, aes_string(x = col)) +
    geom_bar(fill = "skyblue", color = "black") +
    theme_minimal() +
    labs(title = paste("Bar Plot of", col), x = col, y = "Frequency") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))

  # Print the plot
  print(p)
}
```

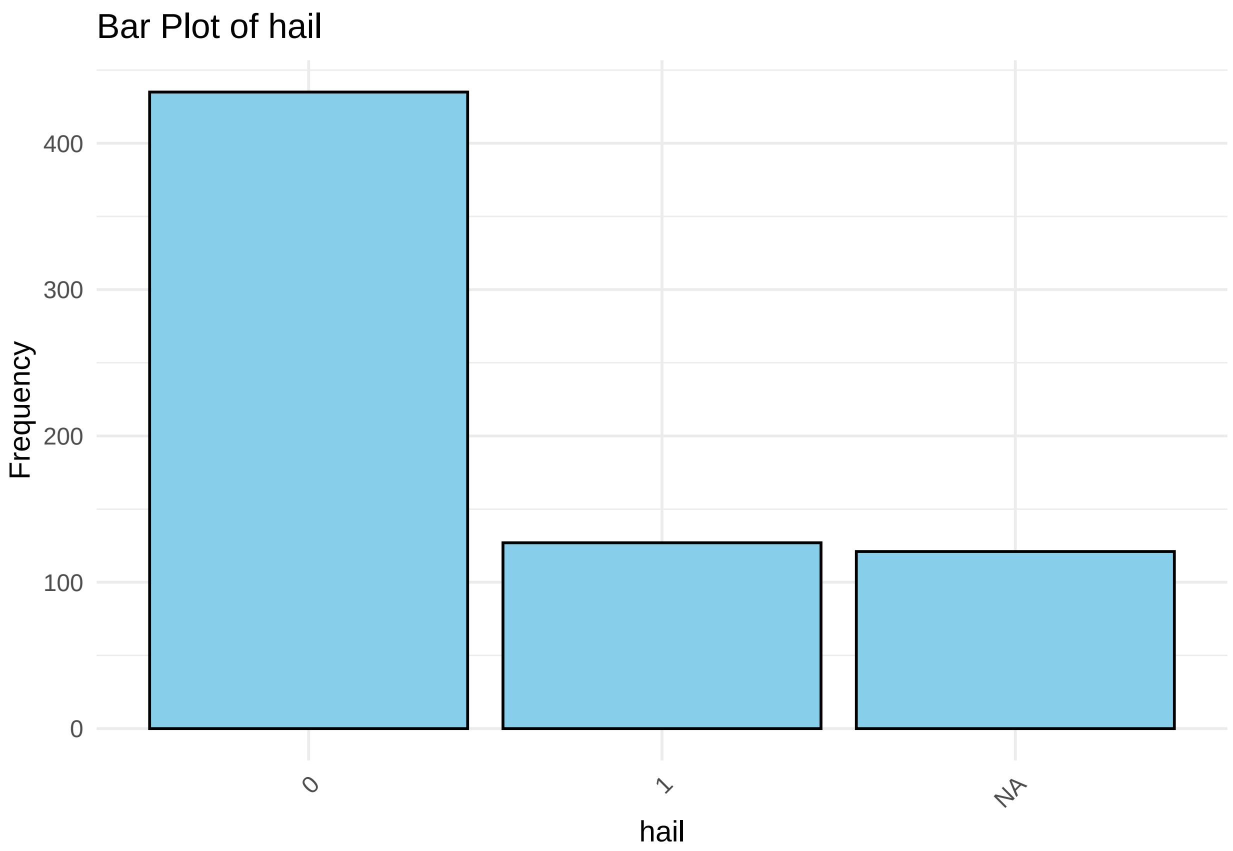

Bar Plot of Class
Frequency
0
25
50
75
2−4−d−injury
alternarialeaf−spot
anthracnose
bacterial−blight
bacterial−pustule
brown−spot
brown−stem−rot
charcoal−rot
cyst−nematode
diaporthe−pod−&−stem−blight
diaporthe−stem−canker
downy−mildew
frog−eye−leaf−spot
herbicide−injury
phyllosticta−leaf−spot
phytophthora−rot
powdery−mildew
purple−seed−stain
rhizoctonia−root−rot
Class

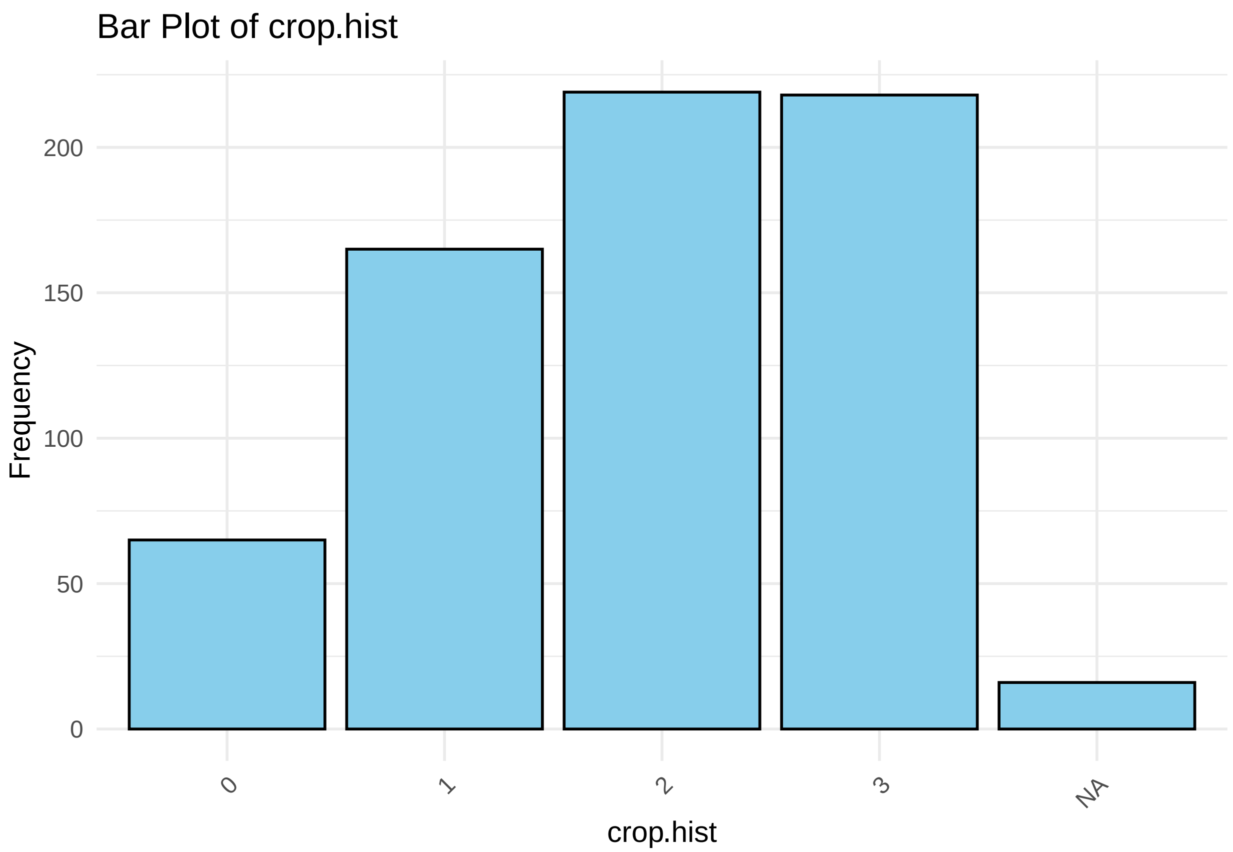

Bar Plot of date
Frequency
150
100
50
0
0
1
2
3
4
5
6
NA
date

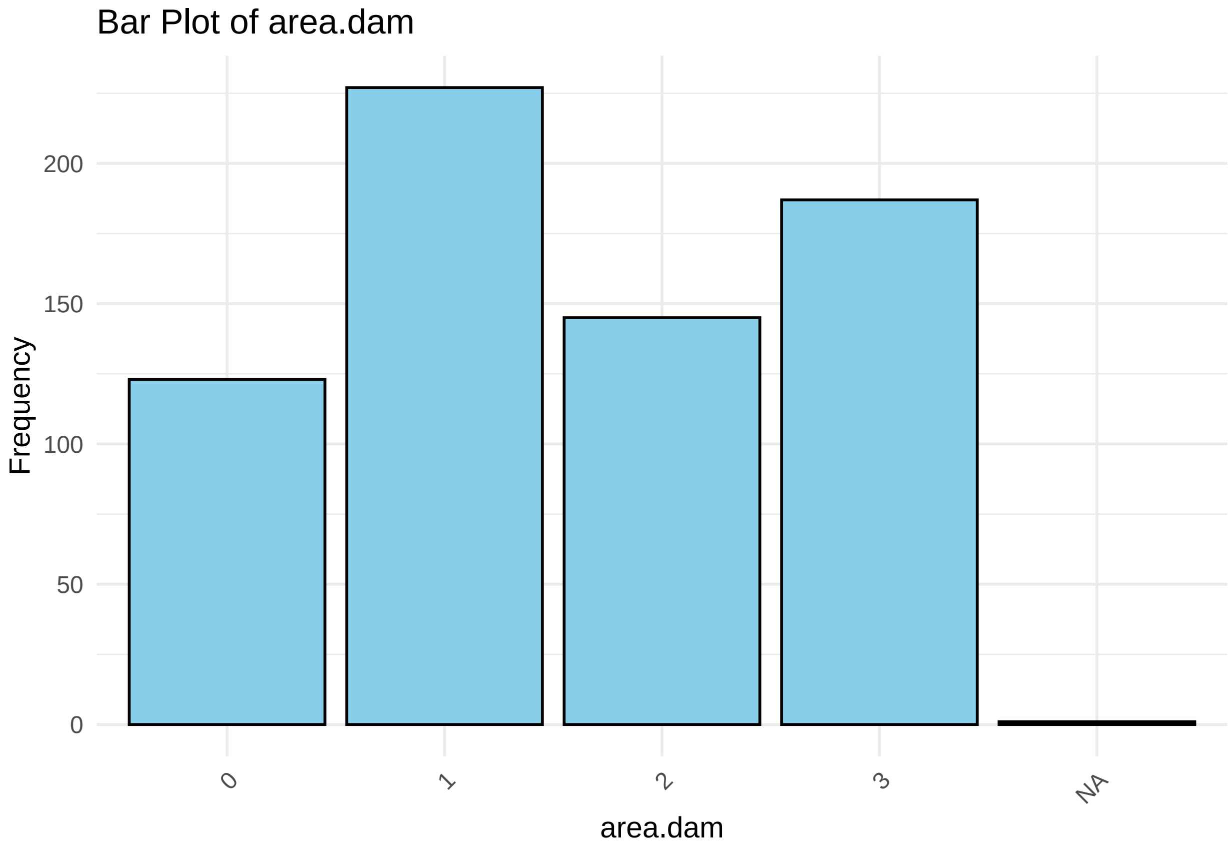

Bar Plot of plant.stand
Frequency
300
200
100
0
0
1
NA
plant.stand

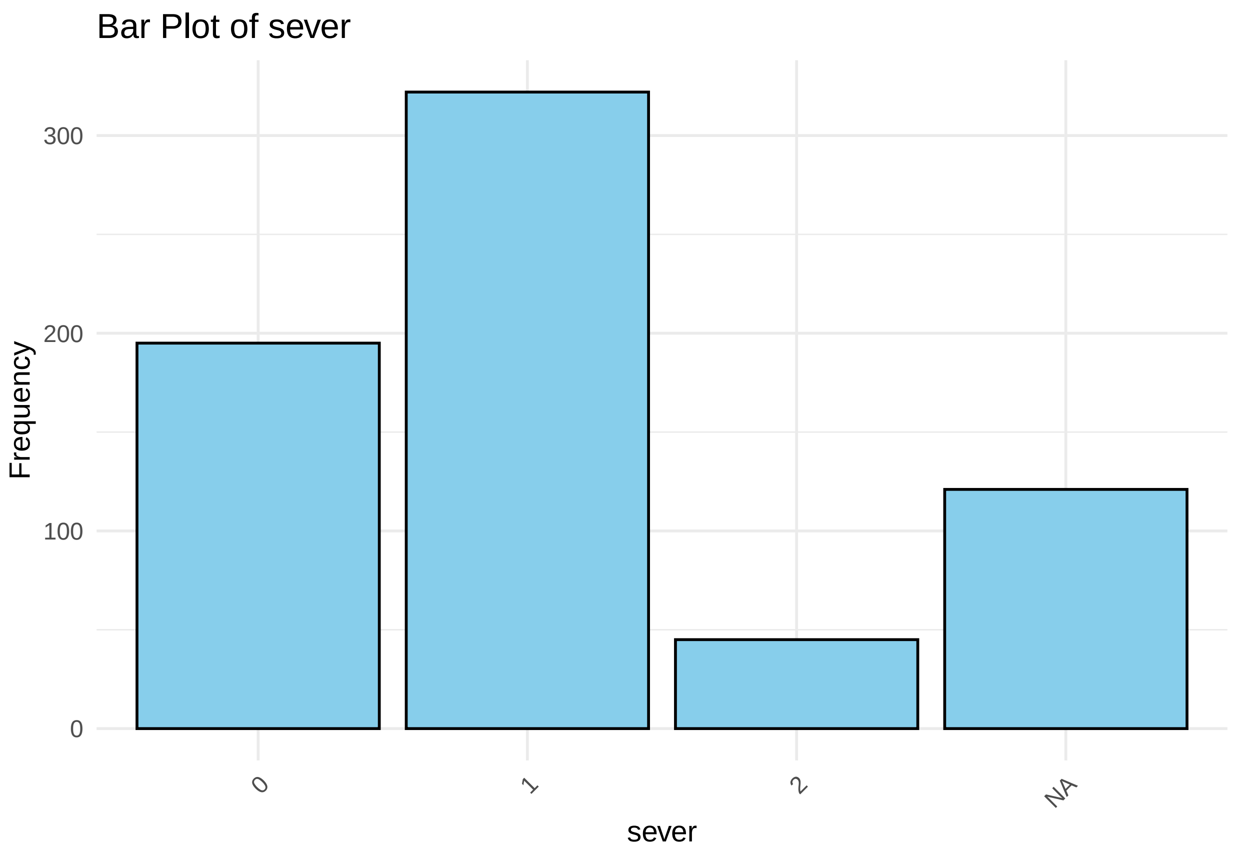

Bar Plot of precip
Frequency
400
300
200
100
0
0
1
2
NA
precip

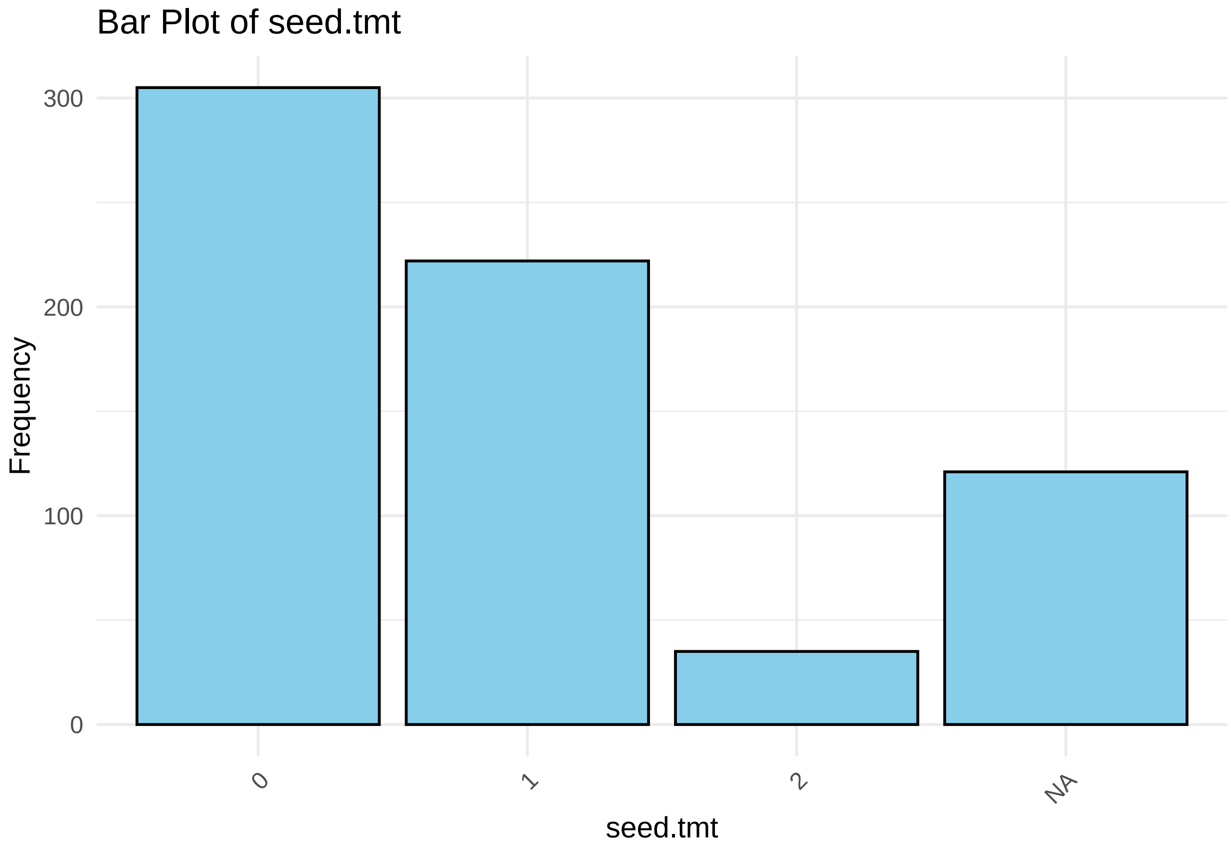

Bar Plot of temp
Frequency
300
200
100
0
0
1
2
NA
temp

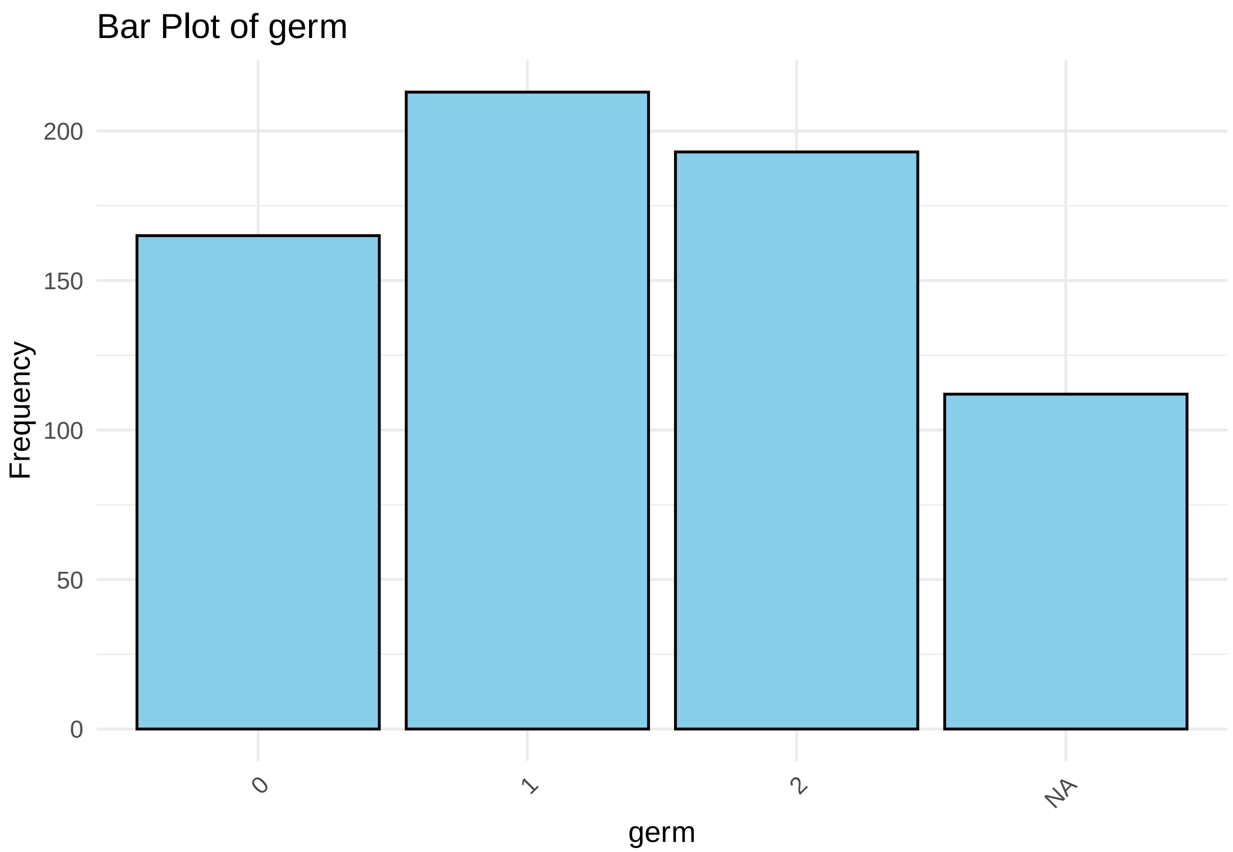

Bar Plot of hail
Frequency
400
300
200
100
0
0
1
NA
hail

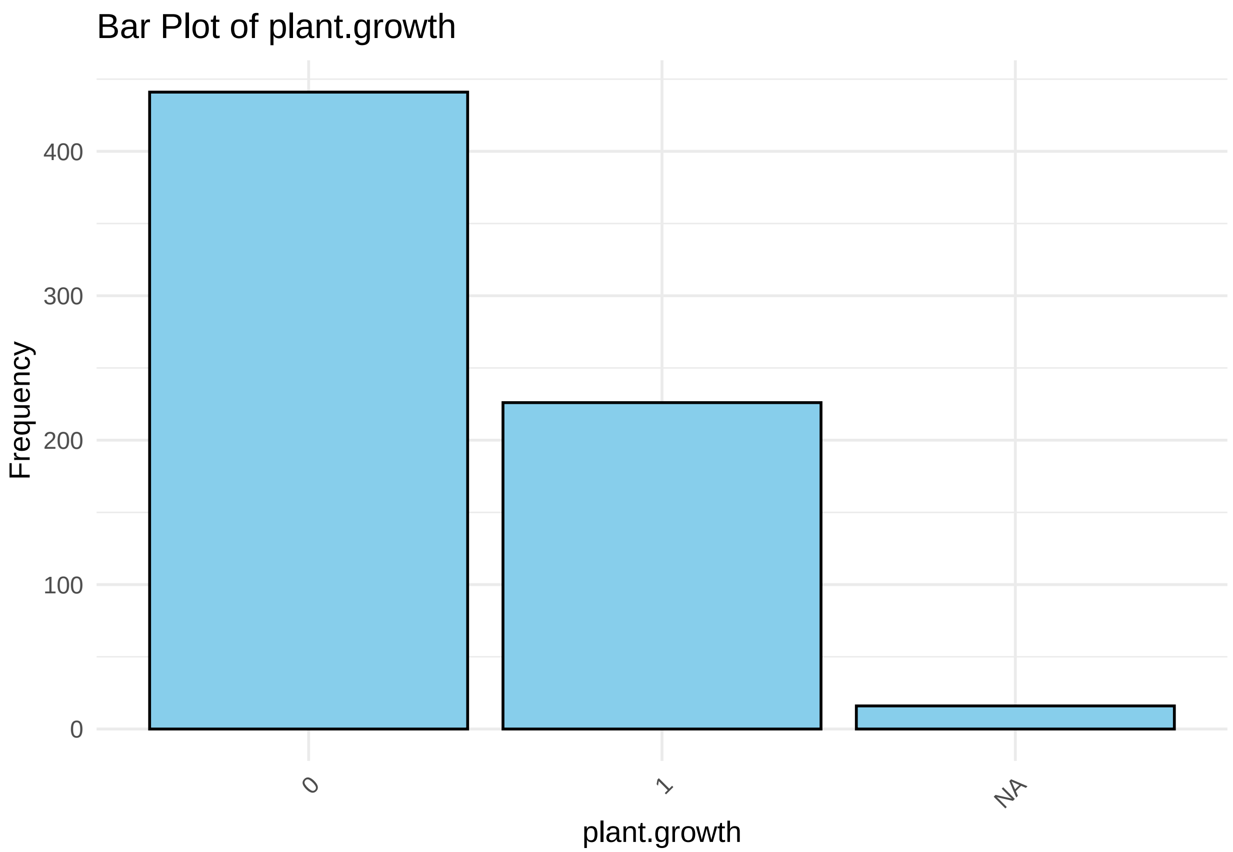

Bar Plot of crop.hist
Frequency
0
50
100
150
200
0
1
2
3
NA
crop.hist

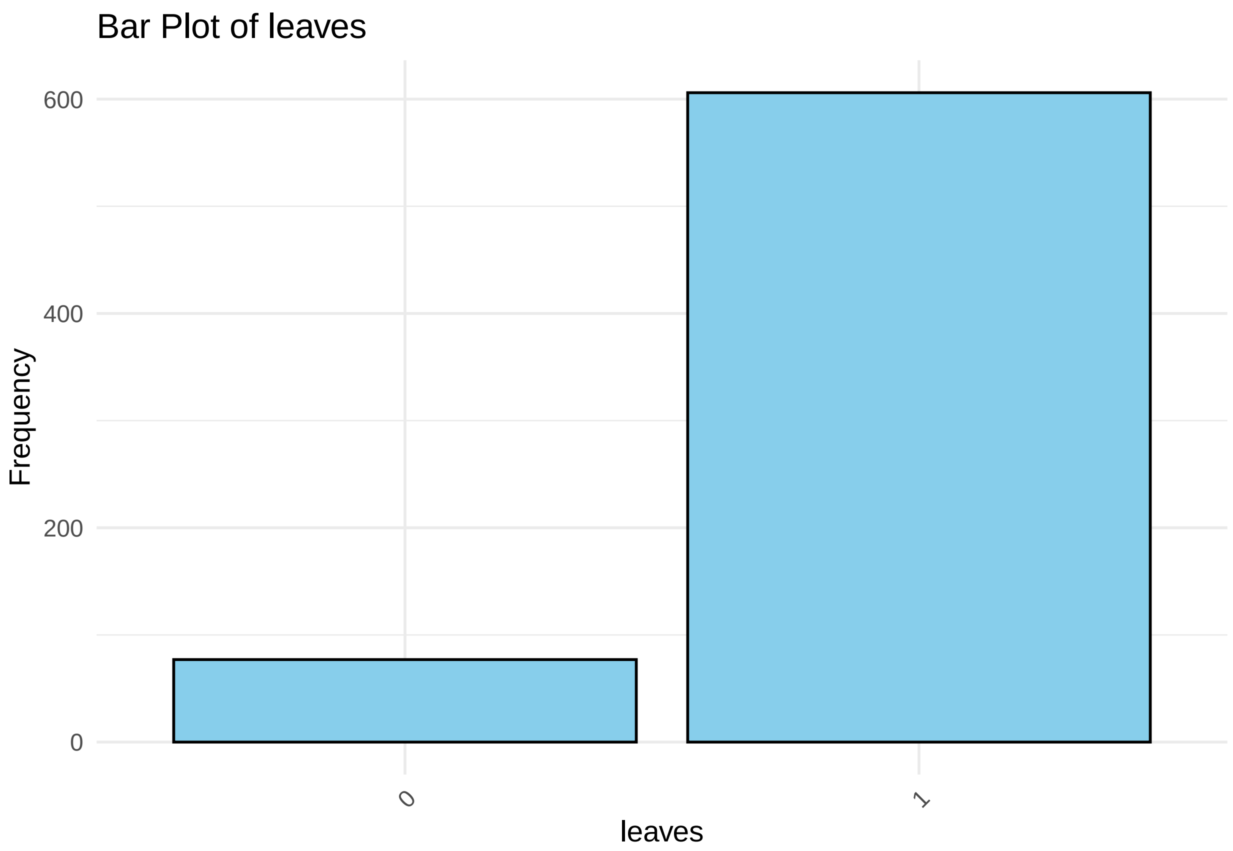

Bar Plot of area.dam
Frequency
200
150
100
50
0
0
1
2
3
NA
area.dam

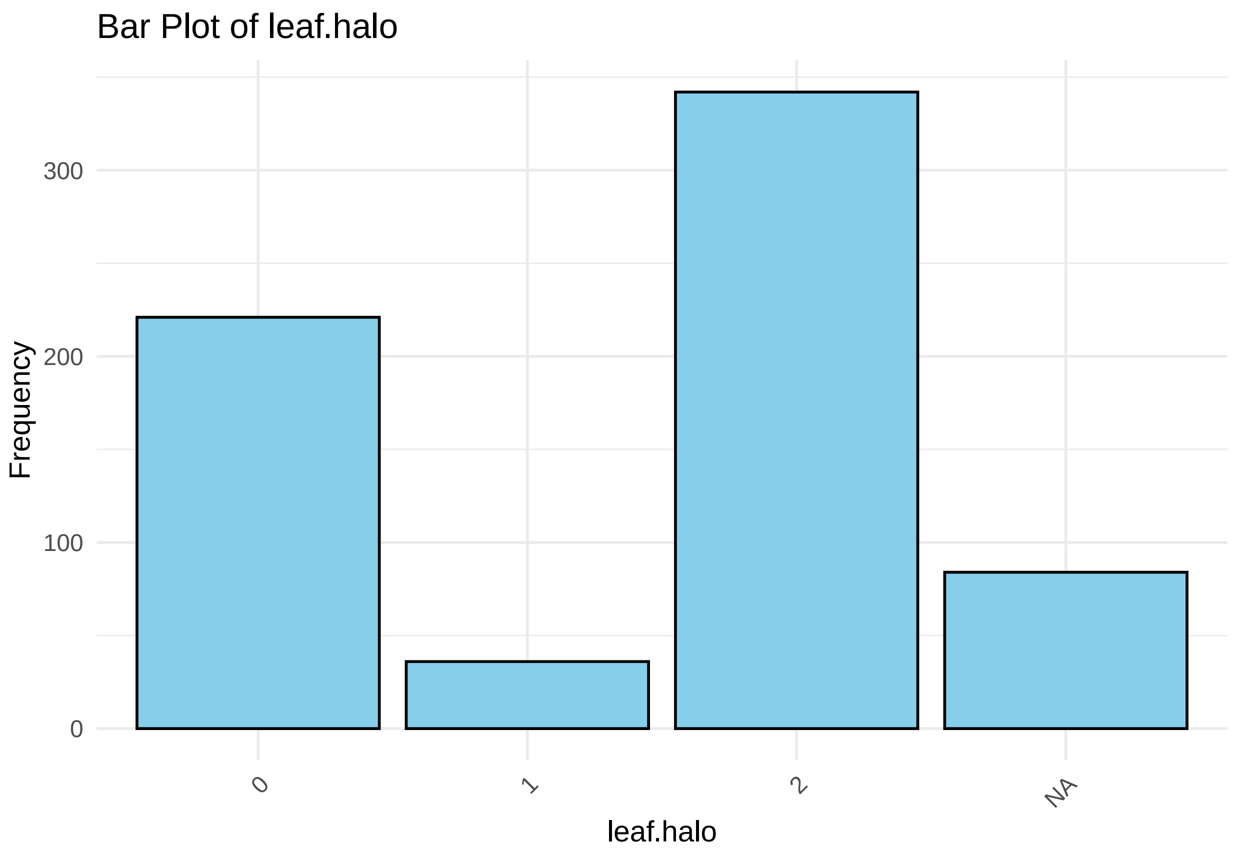

Bar Plot of sever
Frequency
300
200
100
0
0
1
2
NA
sever

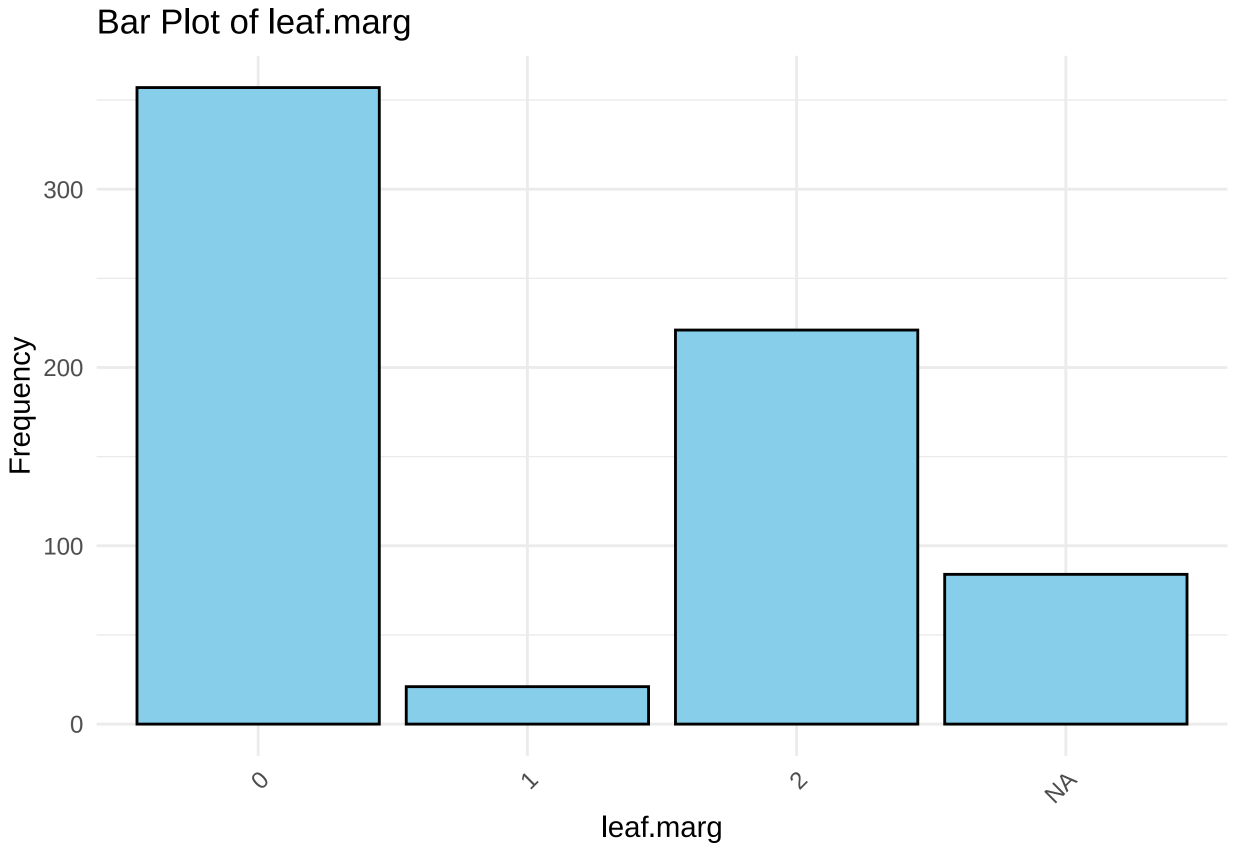

Bar Plot of seed.tmt
Frequency
300
200
100
0
0
1
2
NA
seed.tmt

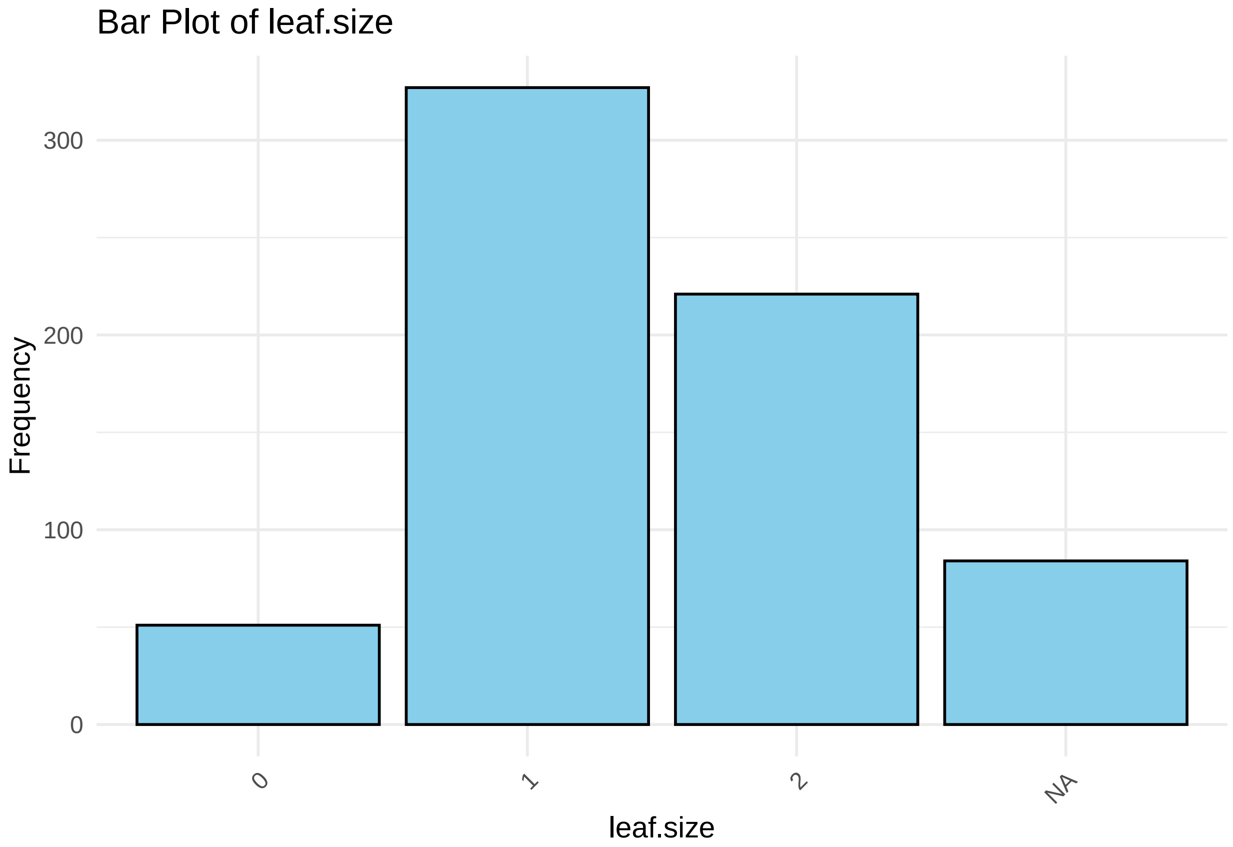

Bar Plot of germ
Frequency
200
150
100
50
0
0
1
2
NA
germ

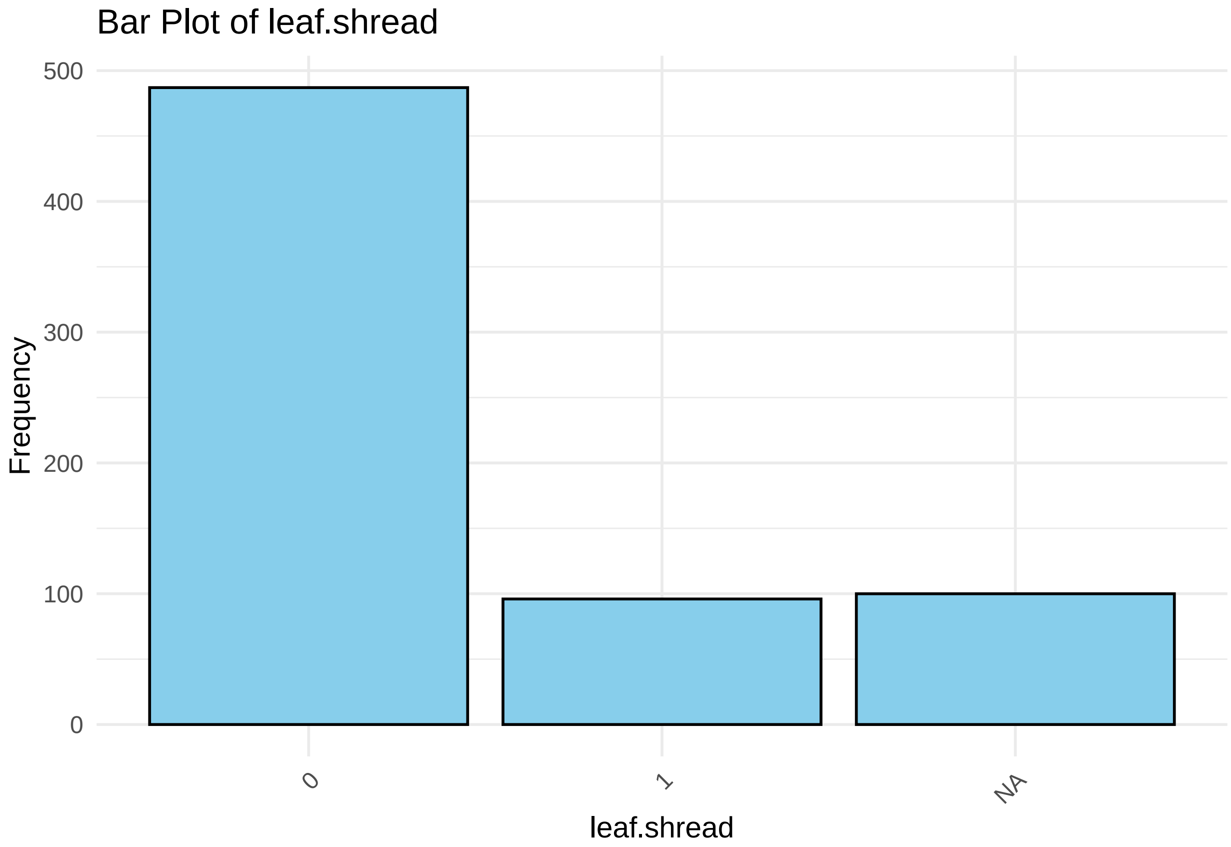

Bar Plot of plant.growth
Frequency
400
300
200
100
0
0
1
NA
plant.growth

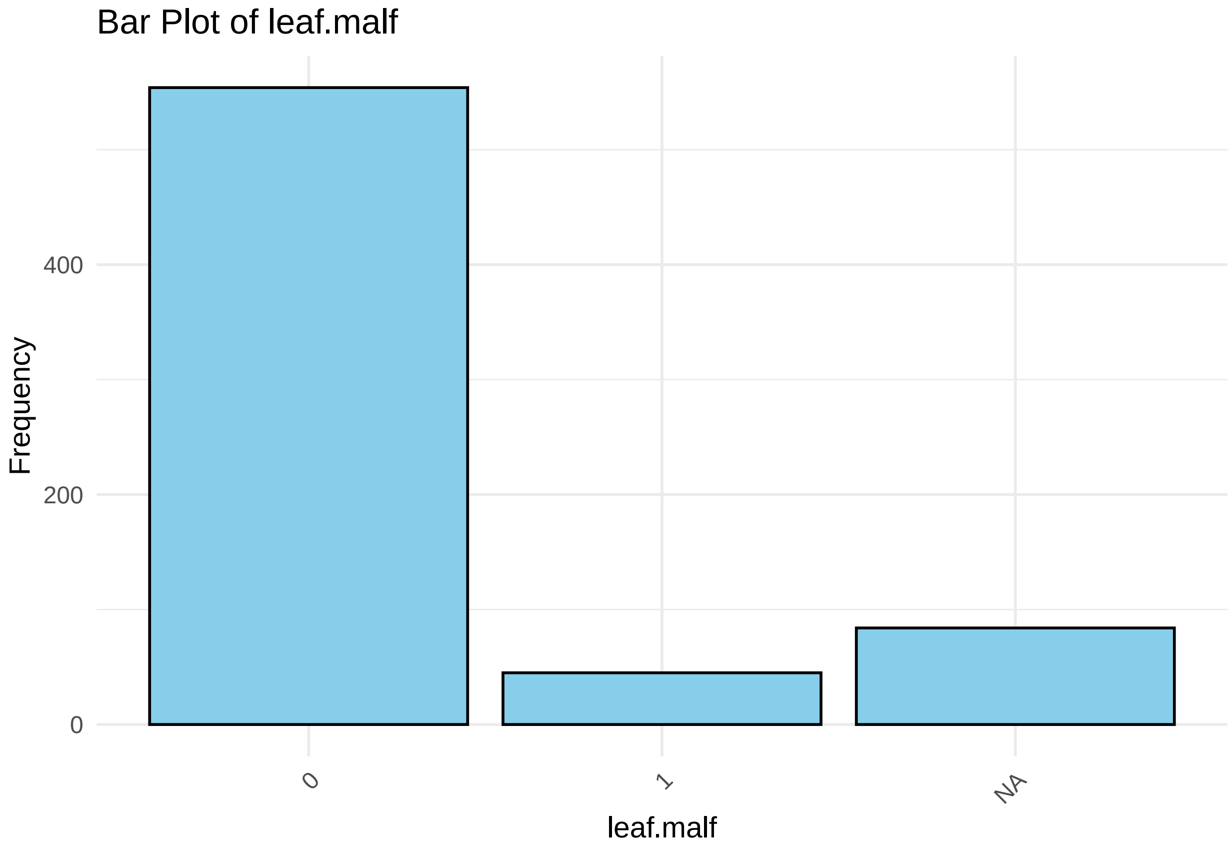

Bar Plot of leaves
Frequency
600
400
200
0
0
1
leaves

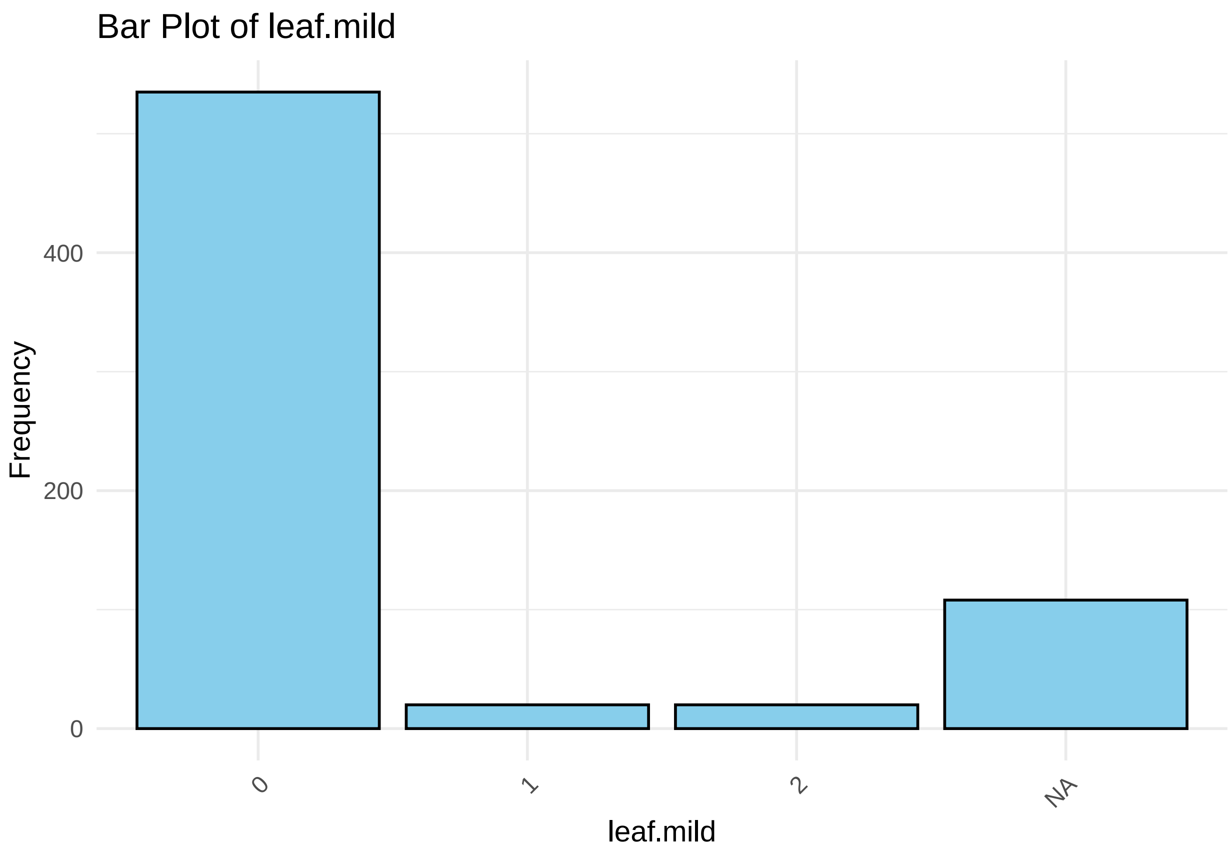

Bar Plot of leaf.halo
Frequency
300
200
100
0
0
1
2
NA
leaf.halo

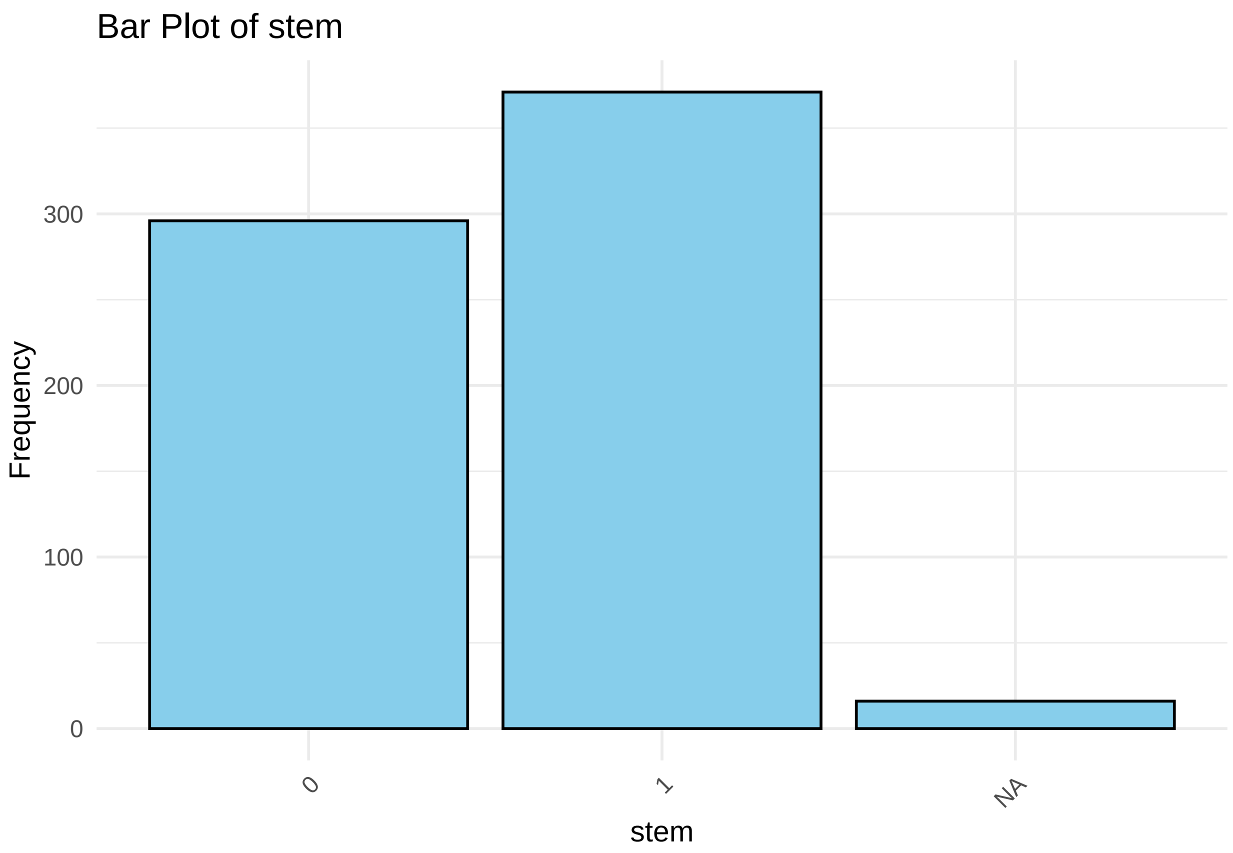

Bar Plot of leaf.marg
Frequency
300
200
100
0
0
1
2
NA
leaf.marg

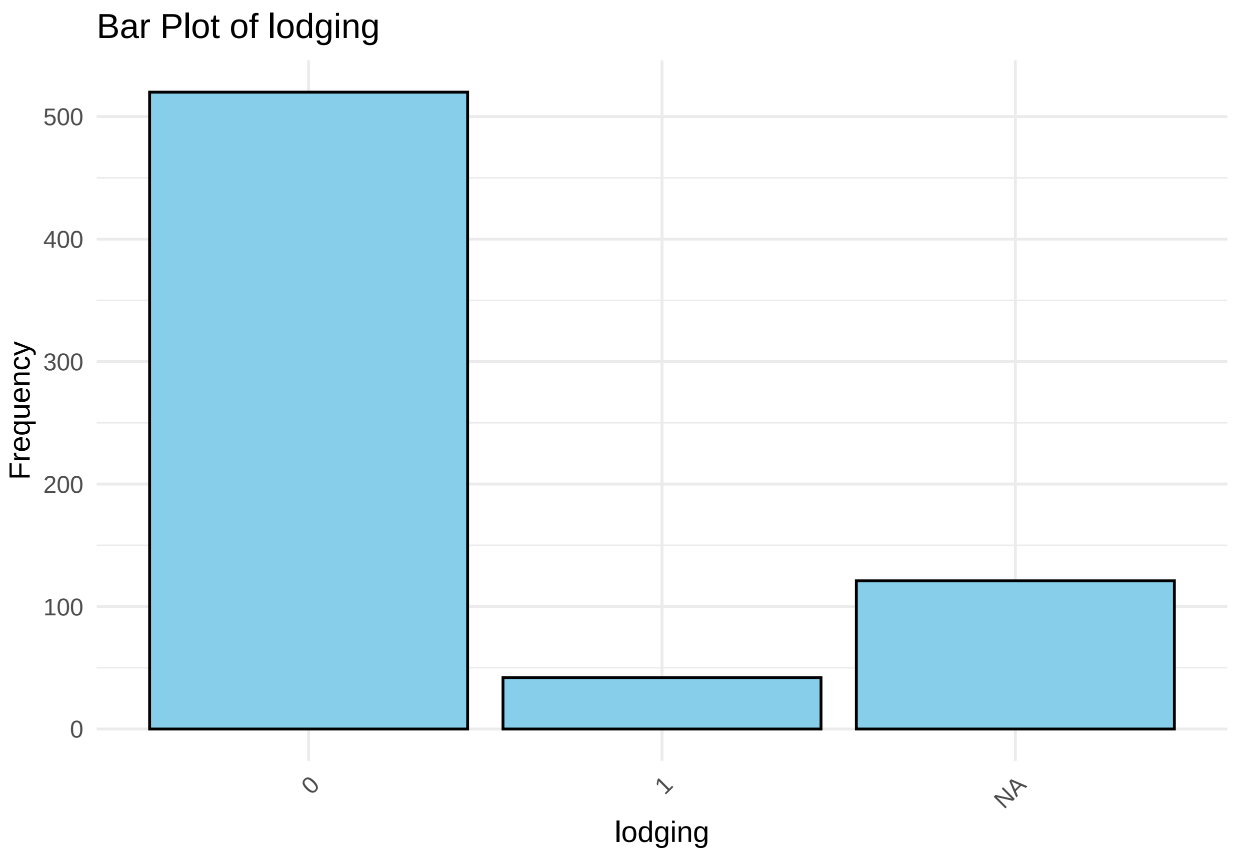

Bar Plot of leaf.size
Frequency
300
200
100
0
0
1
2
NA
leaf.size

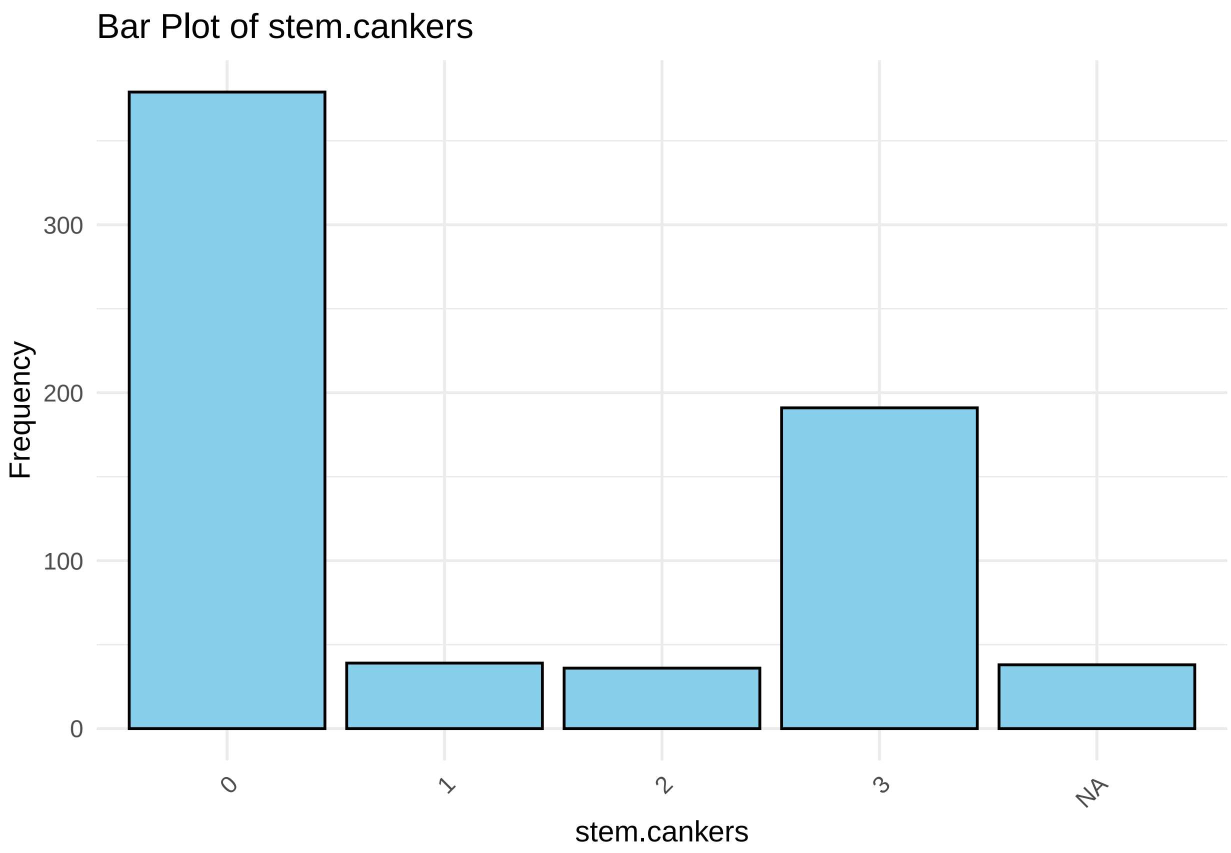

Bar Plot of leaf.shread
Frequency
500
400
300
200
100
0
0
1
NA
leaf.shread

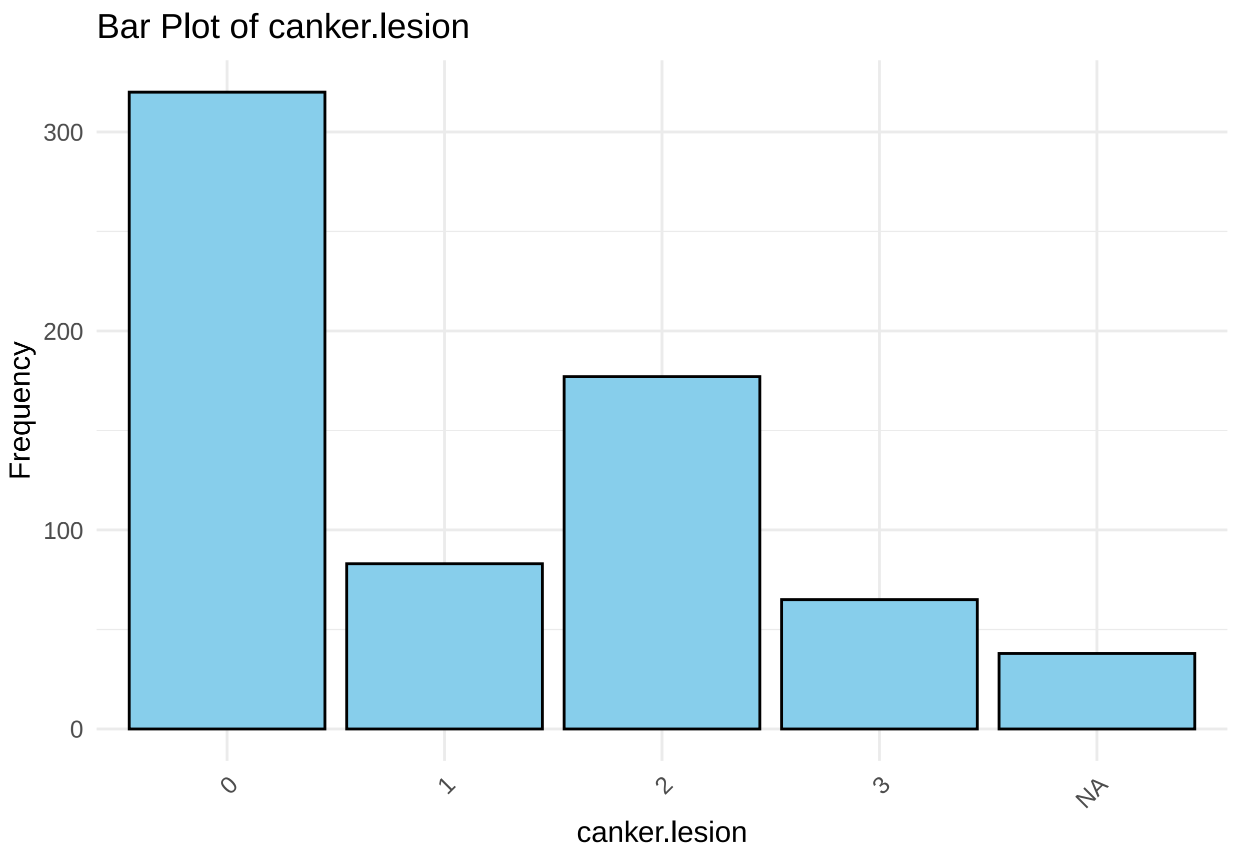

Bar Plot of leaf.malf
Frequency
400
200
0
0
1
NA
leaf.malf

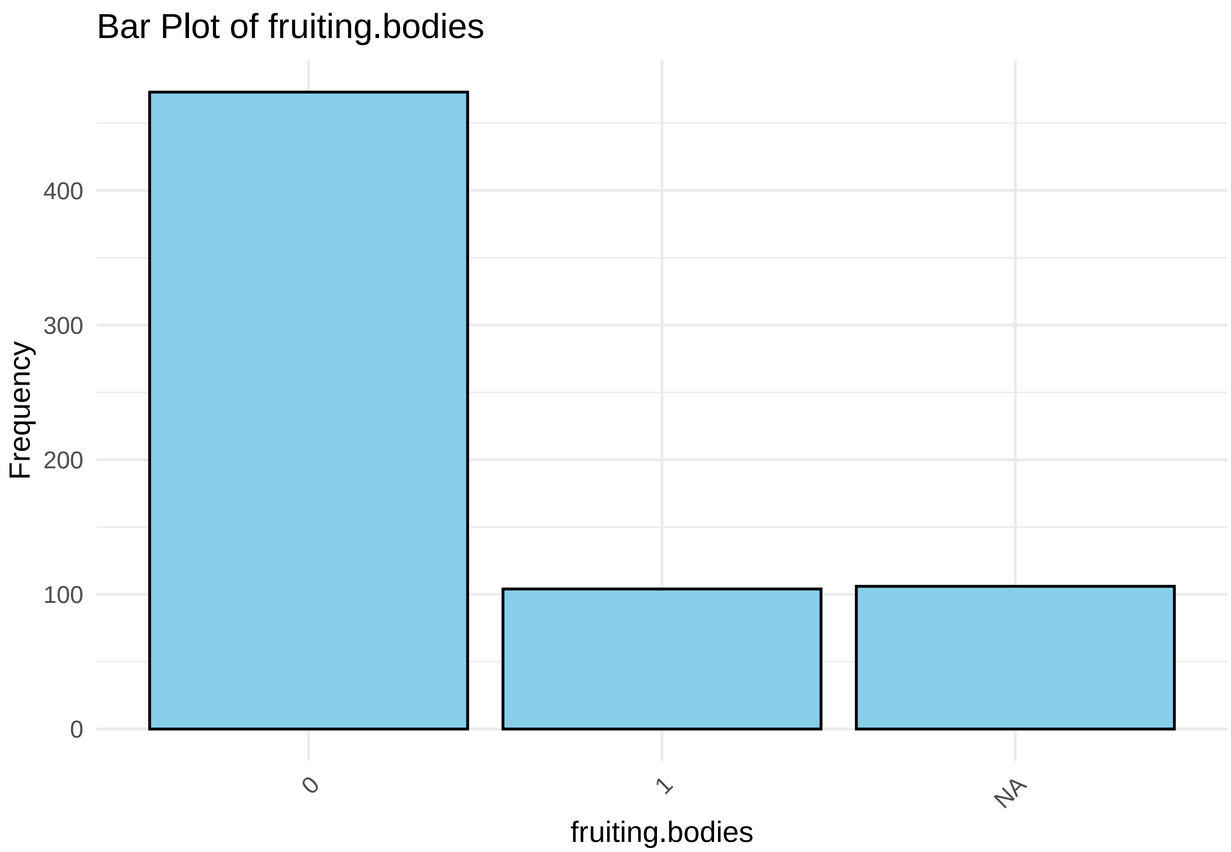

Bar Plot of leaf.mild
Frequency
400
200
0
0
1
2
NA
leaf.mild

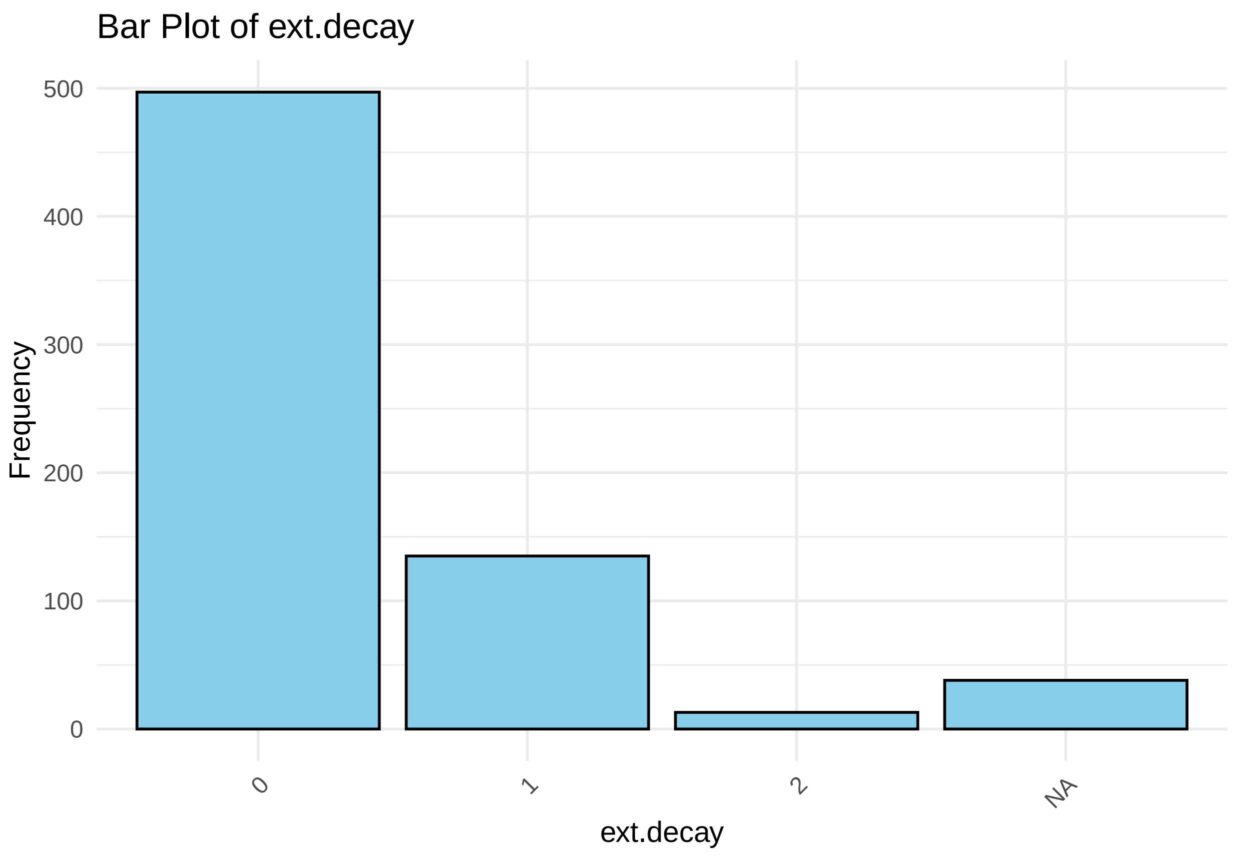

Bar Plot of stem
Frequency
300
200
100
0
0
1
NA
stem

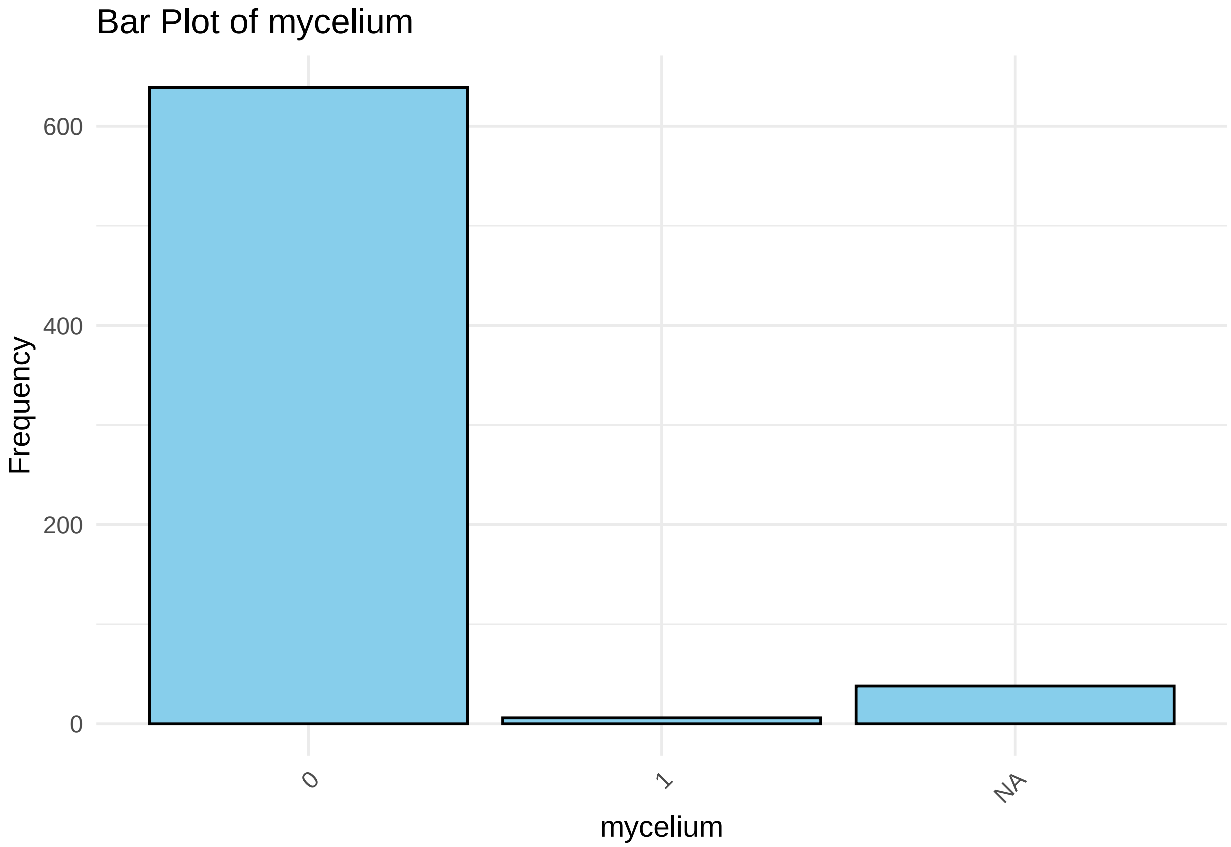

Bar Plot of lodging
Frequency
500
400
300
200
100
0
0
1
NA
lodging

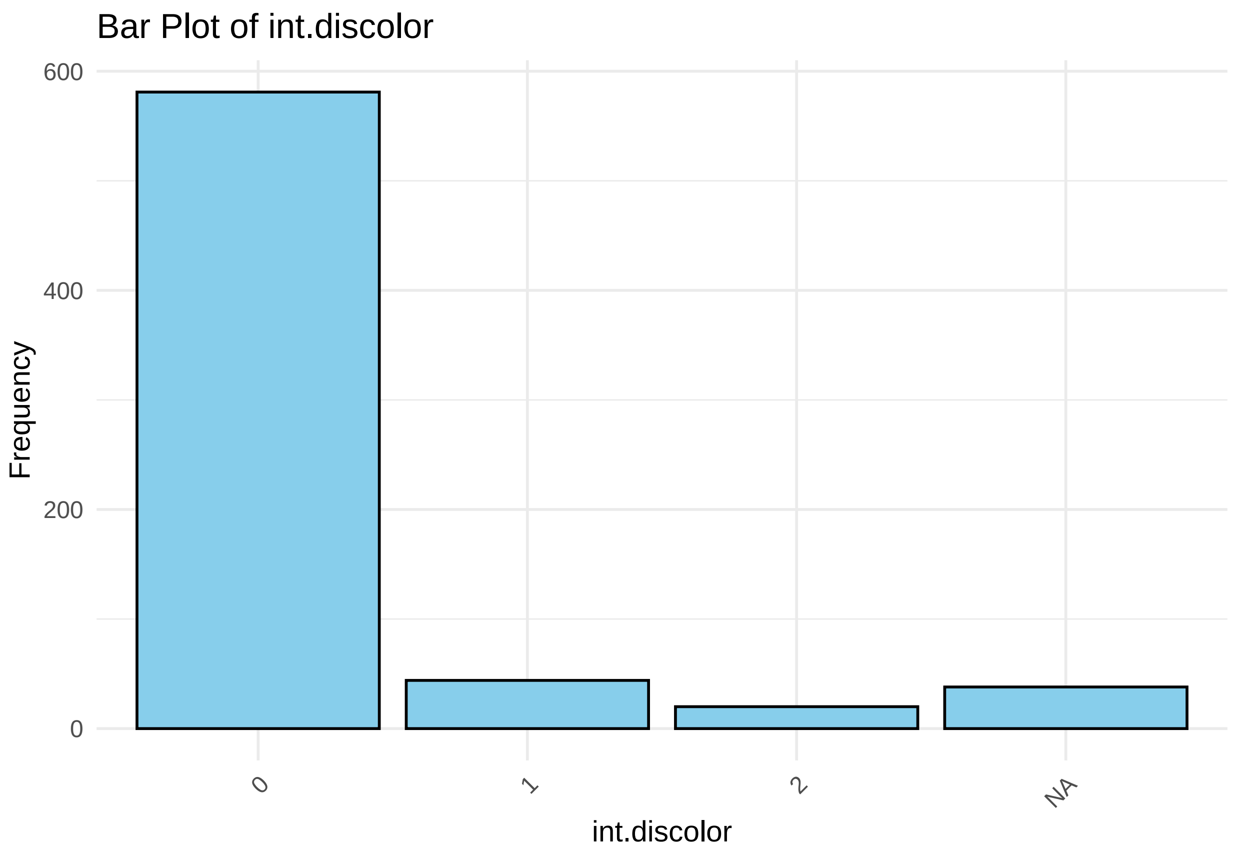

Bar Plot of stem.cankers
Frequency
300
200
100
0
0
1
2
3
NA
stem.cankers

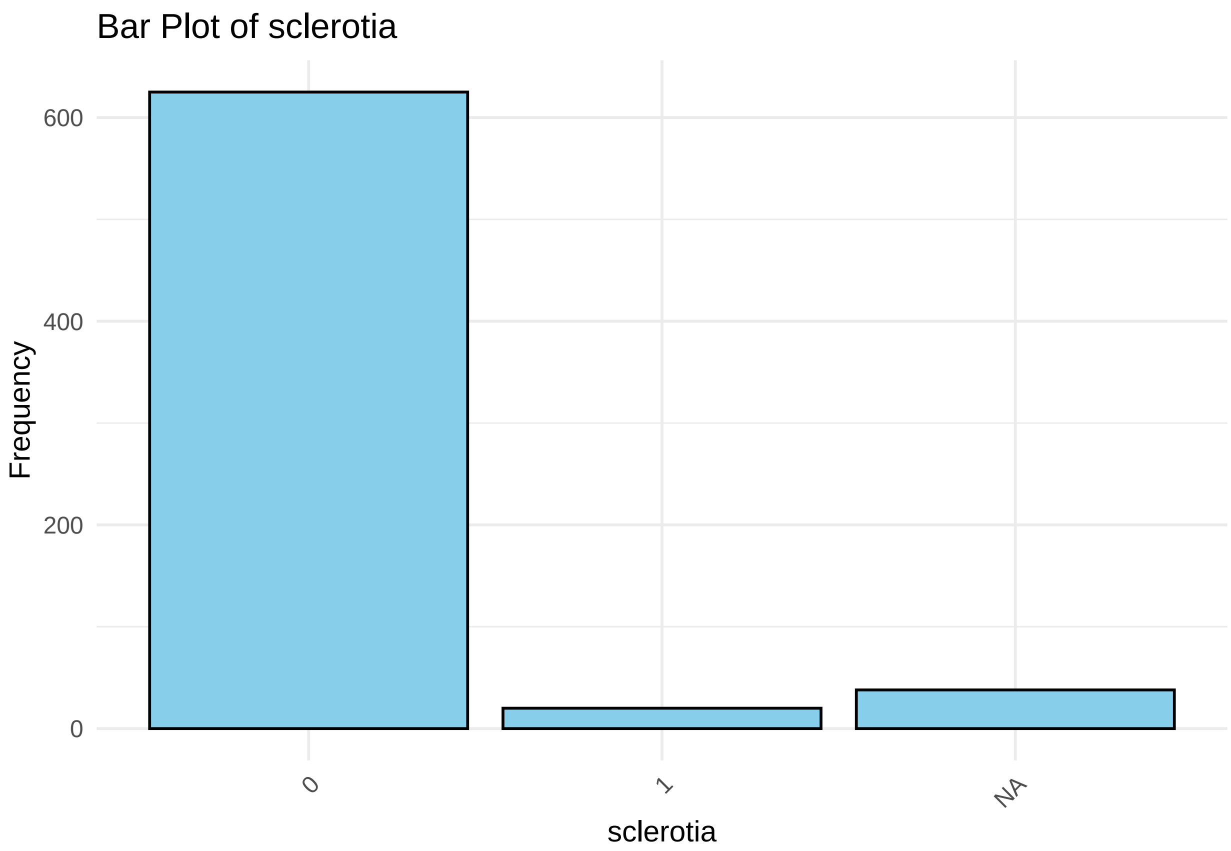

Bar Plot of canker.lesion
Frequency
300
200
100
0
0
1
2
3
NA
canker.lesion

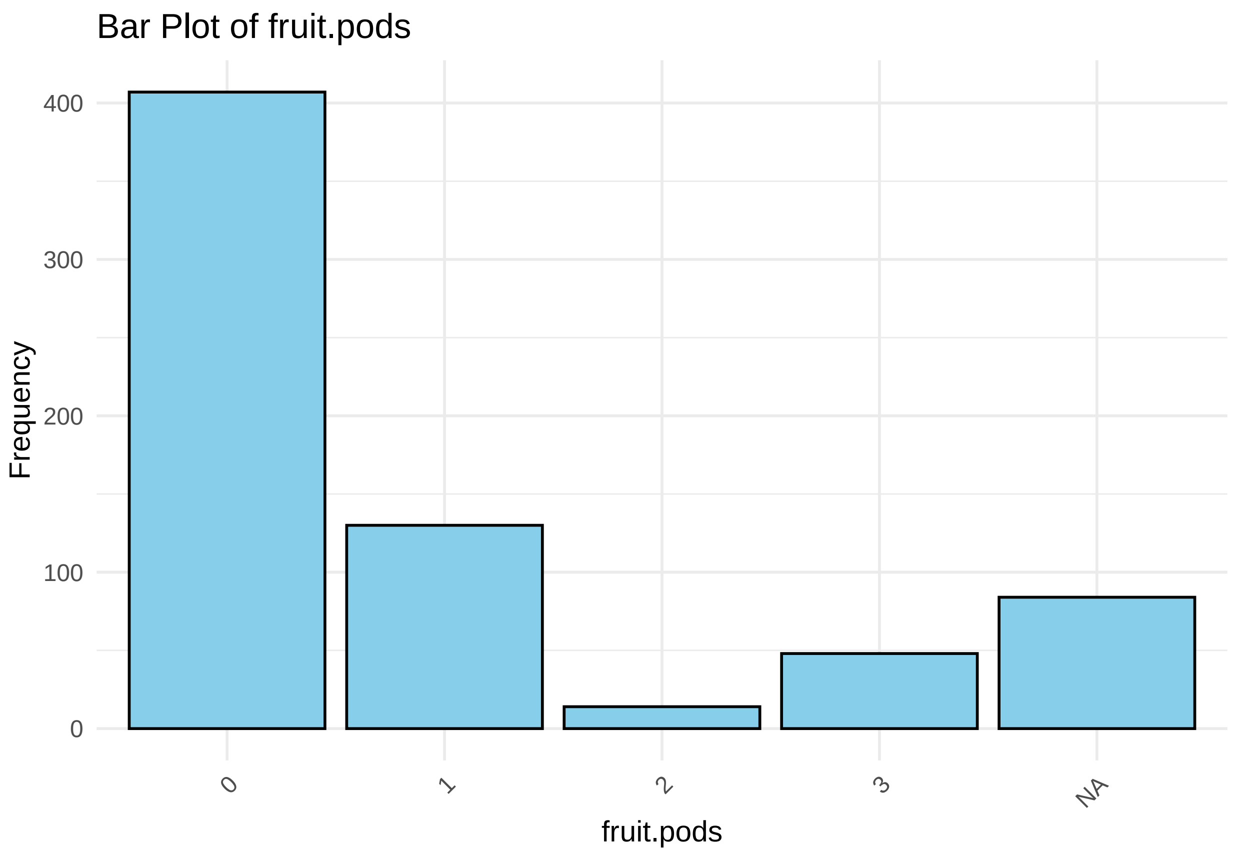

Bar Plot of fruiting.bodies
Frequency
400
300
200
100
0
0
1
NA
fruiting.bodies

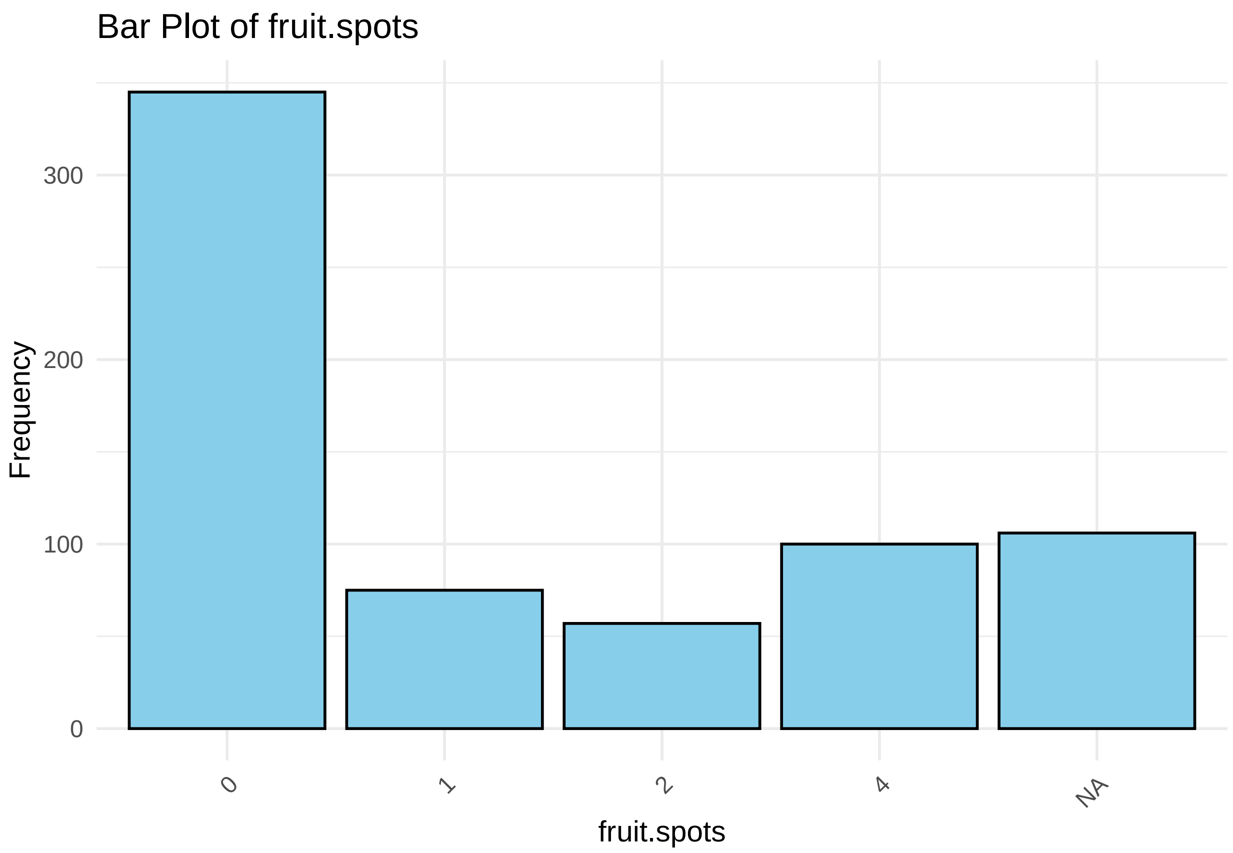

Bar Plot of ext.decay
Frequency
500
400
300
200
100
0
0
1
2
NA
ext.decay

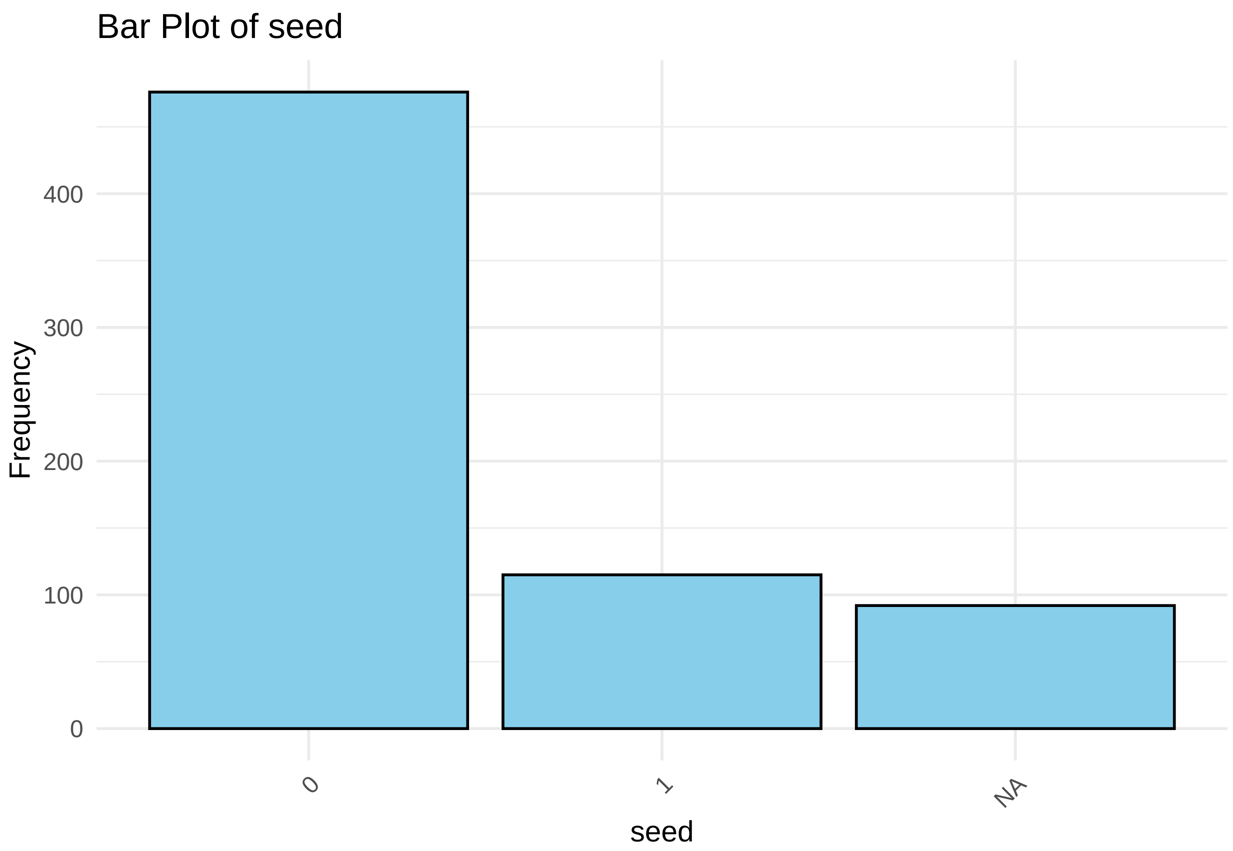

Bar Plot of mycelium
Frequency
600
400
200
0
0
1
NA
mycelium

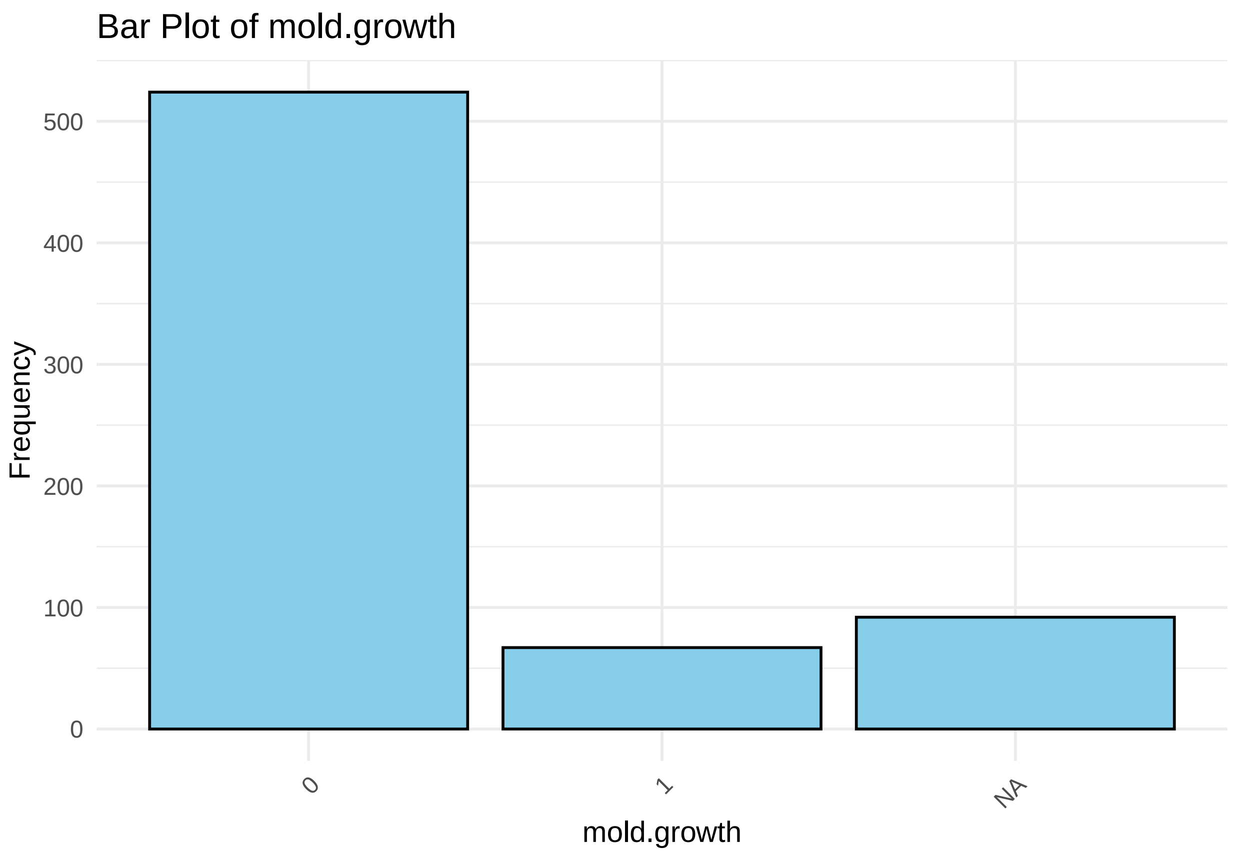

Bar Plot of int.discolor
600
400
200
0
Frequency
0
1
2
NA
int.discolor

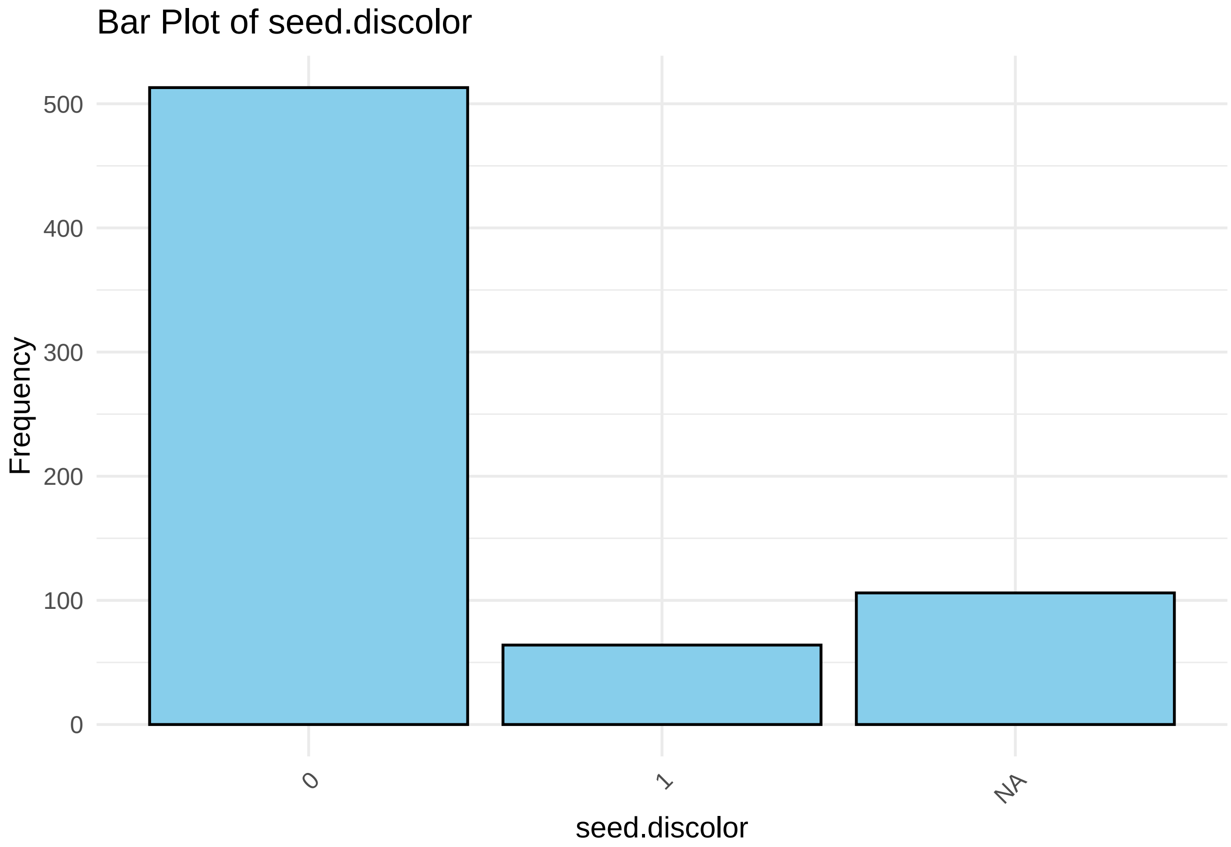

Bar Plot of sclerotia
Frequency
600
400
200
0
0
1
NA
sclerotia

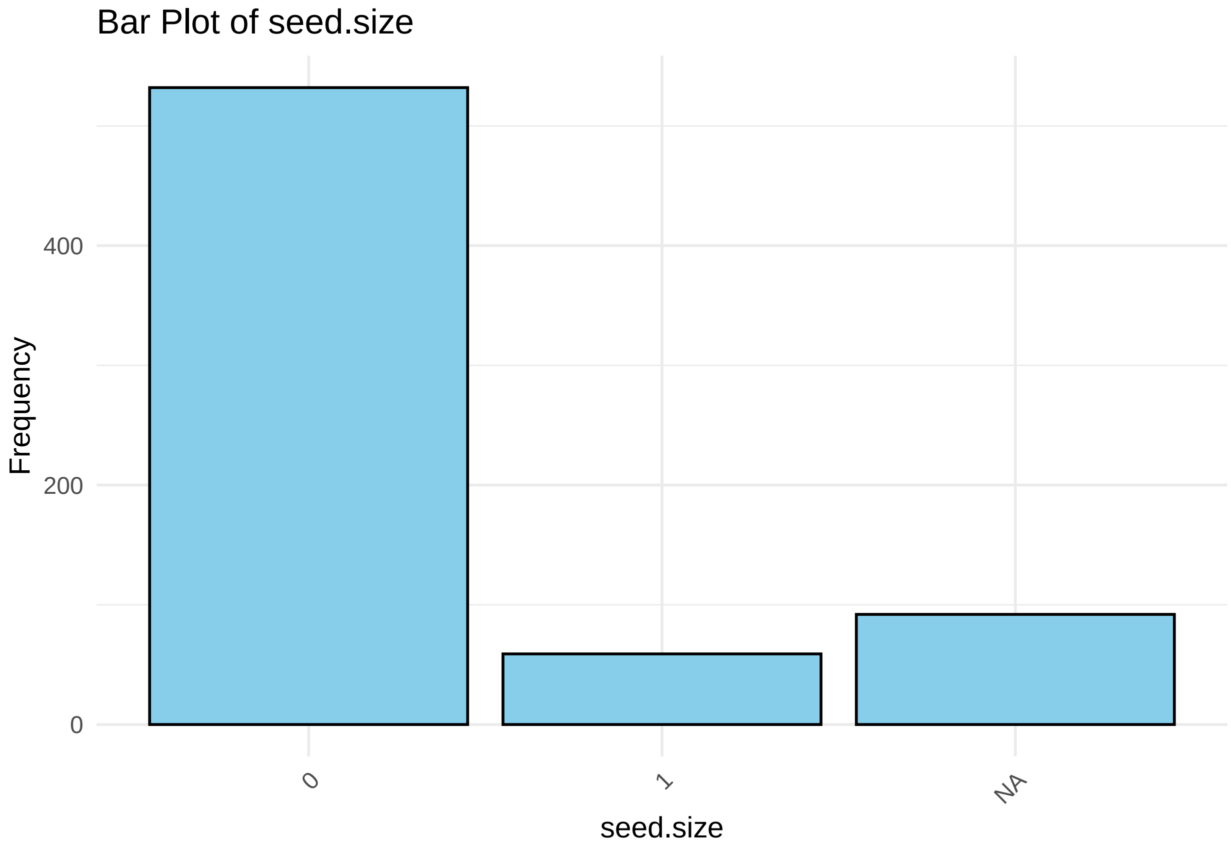

Bar Plot of fruit.pods
Frequency
400
300
200
100
0
0
1
2
3
NA
fruit.pods

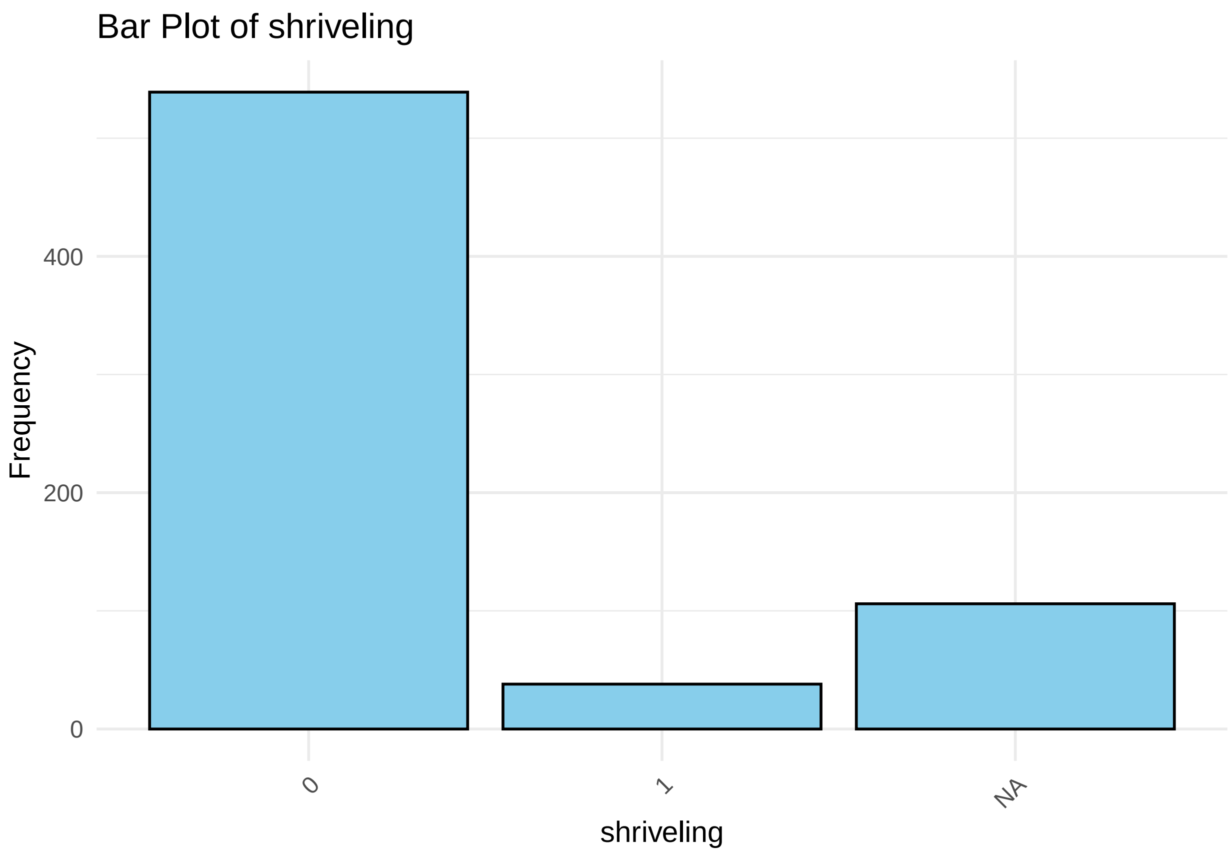

Bar Plot of fruit.spots
Frequency
300
200
100
0
0
1
2
4
NA
fruit.spots

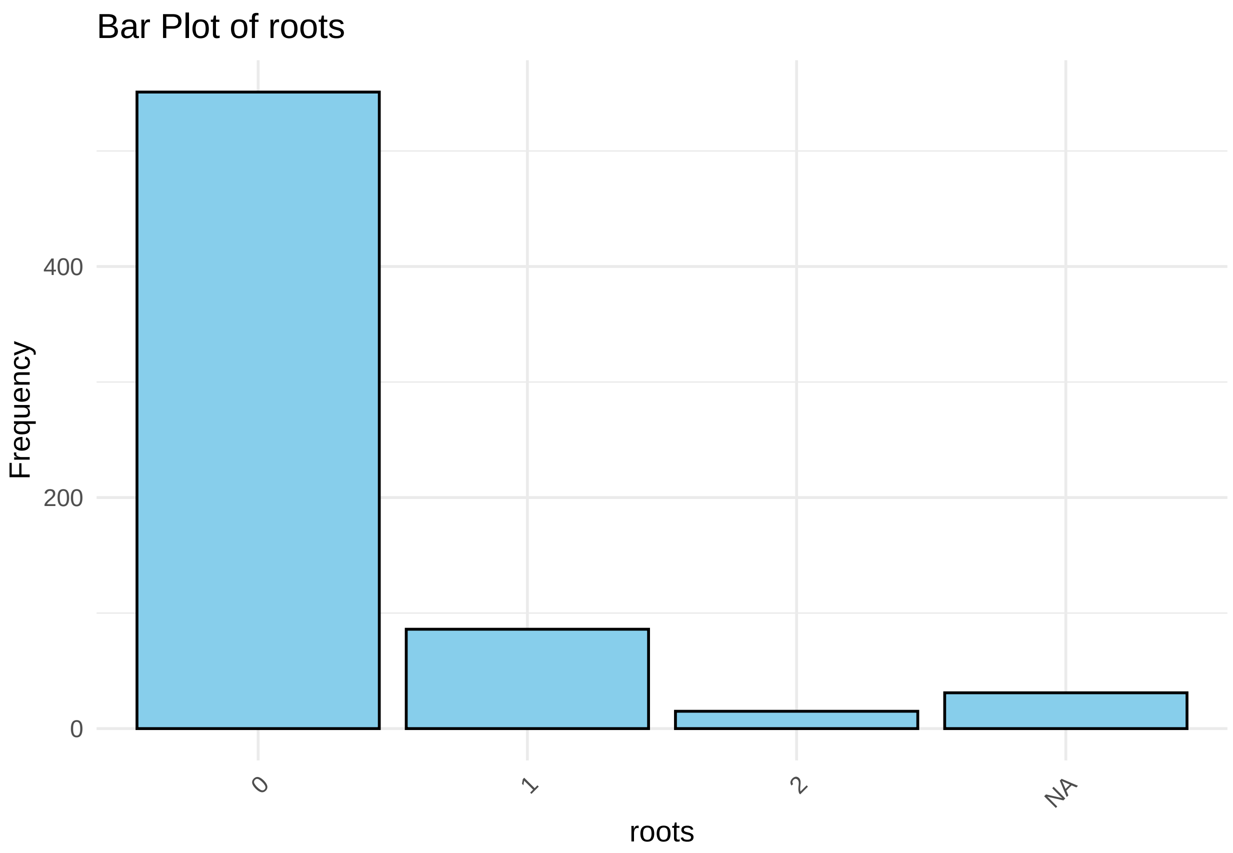

Bar Plot of seed
Frequency
0
100
200
300
400
0
1
NA
seed


Bar Plot of mold.growth
Frequency
500
400
300
200
100
0
0
1
NA
mold.growth


Bar Plot of seed.discolor
Frequency
500
400
300
200
100
0
0
1
NA
seed.discolor


Bar Plot of seed.size
Frequency
400
200
0
0
1
NA
seed.size


Bar Plot of shriveling
Frequency
400
200
0
0
1
NA
shriveling



```r
# Loop through each categorical variable and calculate the proportion of each category
for (col in categorical_columns) {
  cat("Distribution of", col, ":\n")
  print(prop.table(table(Soybean[[col]])))
  cat("\n")
}
```

```
## Distribution of Class :
##
##               2-4-d-injury        alternarialeaf-spot
##                 0.02342606                 0.13323572
##                anthracnose           bacterial-blight
##                 0.06442167                 0.02928258
##          bacterial-pustule                 brown-spot
##                 0.02928258                 0.13469985
##             brown-stem-rot               charcoal-rot
##                 0.06442167                 0.02928258
##              cyst-nematode diaporthe-pod-&-stem-blight
##                 0.02049780                 0.02196193
##       diaporthe-stem-canker               downy-mildew
##                 0.02928258                 0.02928258
##          frog-eye-leaf-spot           herbicide-injury
##                 0.13323572                 0.01171303
##      phyllosticta-leaf-spot             phytophthora-rot
##                 0.02928258                 0.12884334
##             powdery-mildew          purple-seed-stain
##                   0.02928258                   0.02928258
##         rhizoctonia-root-rot
##                   0.02928258
##
## Distribution of date :
##
##          0          1          2          3          4          5          6
## 0.03812317 0.10997067 0.13636364 0.17302053 0.19208211 0.21847507 0.13196481
##
## Distribution of plant.stand :
##
##         0         1
## 0.5471406 0.4528594
##
## Distribution of precip :
##
##         0         1         2
## 0.1147287 0.1736434 0.7116279
##
## Distribution of temp :
##
##         0         1         2
## 0.1225115 0.5727412 0.3047473
##
## Distribution of hail :
##
##         0         1
## 0.7740214 0.2259786
##
## Distribution of crop.hist :
##
##          0          1          2          3
## 0.09745127 0.24737631 0.32833583 0.32683658
##
## Distribution of area.dam :
##
##         0         1         2         3
## 0.1803519 0.3328446 0.2126100 0.2741935
##
## Distribution of sever :
##
##          0          1          2
## 0.34697509 0.57295374 0.08007117
##
## Distribution of seed.tmt :
##
##          0          1          2
## 0.54270463 0.39501779 0.06227758
##
## Distribution of germ :
##
##         0         1         2
## 0.2889667 0.3730298 0.3380035
##
```

```
## Distribution of plant.growth :
##
##         0         1
## 0.6611694 0.3388306
##
## Distribution of leaves :
##
##         0         1
## 0.1127379 0.8872621
##
## Distribution of leaf.halo :
##
##          0          1          2
## 0.36894825 0.06010017 0.57095159
##
## Distribution of leaf.marg :
##
##          0          1          2
## 0.59599332 0.03505843 0.36894825
##
## Distribution of leaf.size :
##
##         0         1         2
## 0.0851419 0.5459098 0.3689482
##
## Distribution of leaf.shread :
##
##         0         1
## 0.8353345 0.1646655
##
## Distribution of leaf.malf :
##
##          0          1
## 0.92487479 0.07512521
##
## Distribution of leaf.mild :
##
##          0          1          2
## 0.93043478 0.03478261 0.03478261
##
## Distribution of stem :
##
##         0         1
## 0.4437781 0.5562219
##
## Distribution of lodging :
##
##         0         1
## 0.9252669 0.0747331
##
## Distribution of stem.cankers :
##
##          0          1          2          3
## 0.58759690 0.06046512 0.05581395 0.29612403
```

```
##
## Distribution of canker.lesion :
##
##         0         1         2         3
## 0.4961240 0.1286822 0.2744186 0.1007752
##
## Distribution of fruiting.bodies :
##
##         0         1
## 0.8197574 0.1802426
##
## Distribution of ext.decay :
##
##          0          1          2
## 0.77054264 0.20930233 0.02015504
##
## Distribution of mycelium :
##
##           0           1
## 0.990697674 0.009302326
##
## Distribution of int.discolor :
##
##          0          1          2
## 0.90077519 0.06821705 0.03100775
##
## Distribution of sclerotia :
##
##          0          1
## 0.96899225 0.03100775
##
## Distribution of fruit.pods :
##
##          0          1          2          3
## 0.67946578 0.21702838 0.02337229 0.08013356
##
## Distribution of fruit.spots :
##
##          0          1          2          4
## 0.59792028 0.12998267 0.09878683 0.17331023
##
## Distribution of seed :
##
##         0         1
## 0.8054146 0.1945854
##
## Distribution of mold.growth :
##
##         0         1
## 0.8866328 0.1133672
##
## Distribution of seed.discolor :
##
##         0         1
## 0.8890815 0.1109185
##
## Distribution of seed.size :
##
##         0         1
## 0.9001692 0.0998308
##
## Distribution of shriveling :
##
##          0          1
## 0.93414211 0.06585789
##
## Distribution of roots :
##
##          0          1          2
## 0.84509202 0.13190184 0.02300613
```

Yes there are highly degenerate variables: Mycelium, Canker Lesion, Sclerotia, Seed Size, Shriveling, Seed Discolor, Mold Growth are highly degenerate because the majority of observations fall into a single category.

The well-distributed variables: Variables like Class (target), Date, Plant Stand, and Precip show more balance and provide better variability for modeling.

(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?

```
# Check overall proportion of missing data
total_missing <- sum(is.na(Soybean)) / (nrow(Soybean) * ncol(Soybean)) * 100
cat("Overall percentage of missing data: ", total_missing, "%\n\n")
```

```
## Overall percentage of missing data:  9.504636 %
```

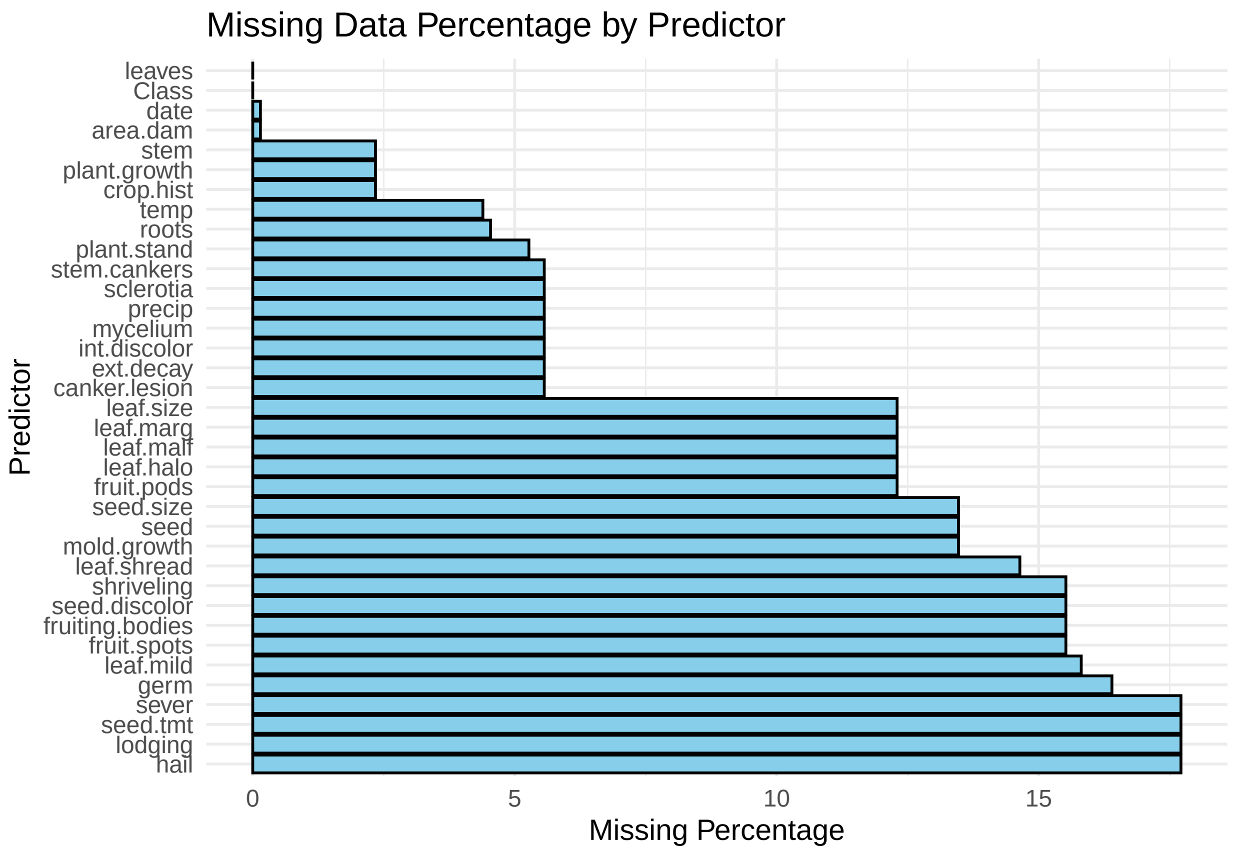
```
# Check missing data for each predictor
missing_data_summary <- colSums(is.na(Soybean)) / nrow(Soybean) * 100
cat("Missing data percentage for each predictor:\n")
```

```
## Missing data percentage for each predictor:
```

```
print(missing_data_summary)
```

```
##           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
```

```
# Visualize missing data distribution across predictors
missing_data_df <- data.frame(
  Predictor = names(missing_data_summary),
  MissingPercentage = missing_data_summary
)

# Plot bar chart of missing data percentage for each predictor
ggplot(missing_data_df, aes(x = reorder(Predictor, -MissingPercentage), y = MissingPercentage)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  coord_flip() +
  labs(title = "Missing Data Percentage by Predictor", x = "Predictor", y = "Missing Percentage") +
  theme_minimal()
```



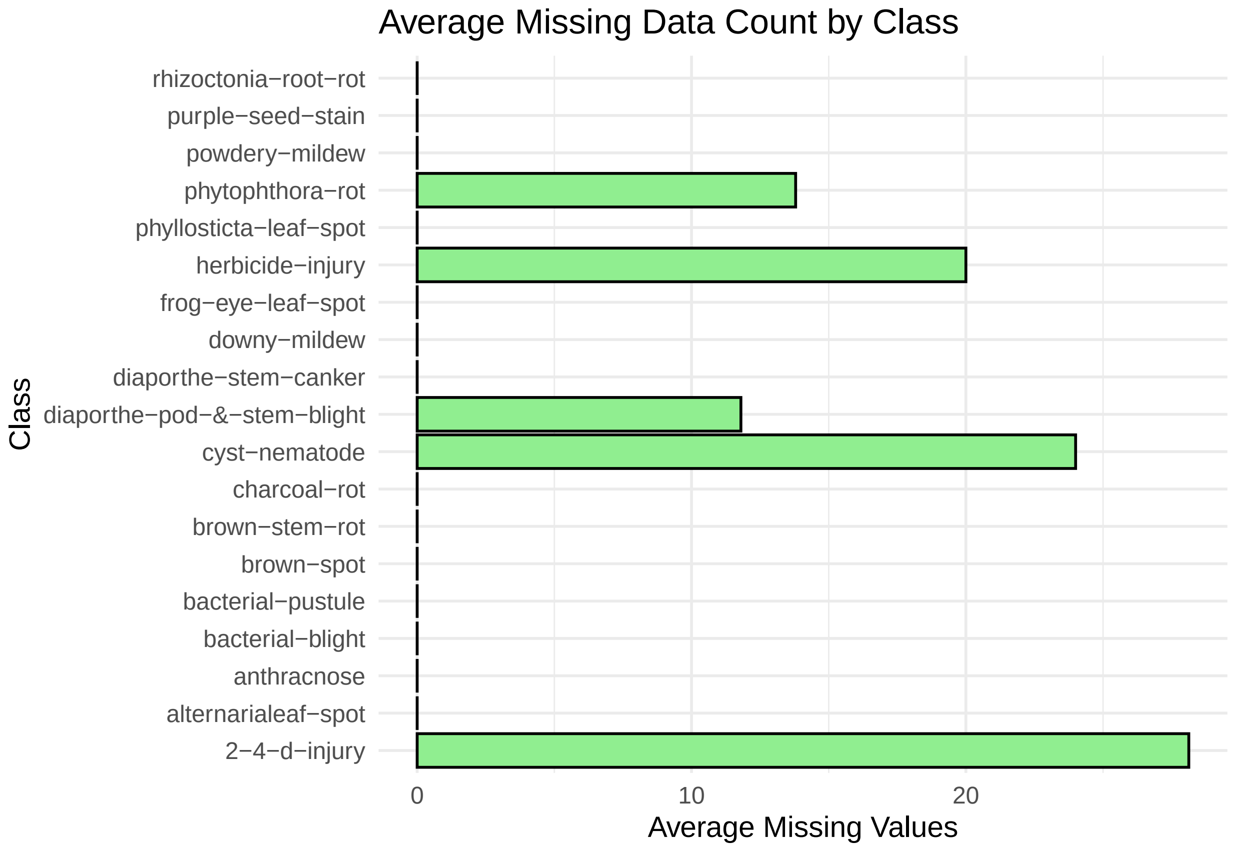
```
# Investigate if the pattern of missing data is related to classes
missing_by_class <- Soybean %>%
  mutate(MissingCount = rowSums(is.na(Soybean))) %>%
  group_by(Class) %>%
  summarise(AvgMissing = mean(MissingCount))

# Print missing data summary by class
cat("\nAverage missing data count by class:\n")
```

```
##
## Average missing data count by class:
```

```
print(missing_by_class)
```

```
## # A tibble: 19 x 2
##    Class                       AvgMissing
##    <fct>                            <dbl>
##  1 2-4-d-injury                      28.1
##  2 alternarialeaf-spot                0  
##  3 anthracnose                        0  
##  4 bacterial-blight                   0  
##  5 bacterial-pustule                  0  
##  6 brown-spot                         0  
##  7 brown-stem-rot                     0  
##  8 charcoal-rot                       0  
##  9 cyst-nematode                     24  
## 10 diaporthe-pod-&-stem-blight       11.8
## 11 diaporthe-stem-canker              0  
## 12 downy-mildew                       0  
## 13 frog-eye-leaf-spot                 0  
## 14 herbicide-injury                  20  
## 15 phyllosticta-leaf-spot             0  
## 16 phytophthora-rot                  13.8
## 17 powdery-mildew                     0  
## 18 purple-seed-stain                  0  
## 19 rhizoctonia-root-rot               0  
```

```
# Visualize missing data pattern by class
ggplot(missing_by_class, aes(x = Class, y = AvgMissing)) +
  geom_bar(stat = "identity", fill = "lightgreen", color = "black") +
  coord_flip() +
  labs(title = "Average Missing Data Count by Class", x = "Class", y = "Average Missing Values") +
  theme_minimal()
```



Yes, some predictors are significantly more likely to have missing data compared to others. According to the bar plot and data you've shared:

Hail, Sever, Seed Treatment (seed.tmt), Germ, Leaf Mildness (leaf.mild), and Shriveling have the highest percentages of missing data, each with more than 15% missing values. Hail, Sever, Seed Treatment, Lodging all have about 17.7% missing data. Variables like Leaves, Class, Date, and Area Damage have little to no missing data, making them more reliable predictors.

Hail, Sever, Seed Treatment, Germ, Leaf Halo, Leaf Shread, Leaf Malformation (leaf.malf) all exhibit significantly high percentages of missing data. These predictors might need special handling, such as imputation or exclusion, depending on their relevance to the analysis.

Predictors most likely to have missing data: Hail, Sever, Seed Treatment, Germ, Leaf Mildness, and Shriveling. These variables should be carefully considered in the analysis, as their high rates of missingness could influence model performance.

Yes, the pattern of missing data appears to vary by class. From the second bar plot:

2-4-D-Injury and Cyst-Nematode classes show the highest average missing values, with 28.12% and 24.00% missing data on average, respectively. Other classes, such as Anthracnose, Bacterial Blight, Bacterial Pustule, Brown Spot, and Charcoal Rot, show no missing data or very little missing data. Some classes (e.g., Diaporthe Pod & Stem Blight) exhibit moderate amounts of missing data (~11.8%).

2-4-D-Injury and Cyst-Nematode: These classes have the most missing data, with more than 20% of their values missing on average. This could potentially affect the model's ability to classify these diseases accurately. Classes with little or no missing data, such as Anthracnose and Bacterial Blight, will likely not be affected by missingness.

Classes affected by missing data: The missing data appears to disproportionately affect certain classes, particularly 2-4-D-Injury and Cyst-Nematode, which show significantly higher rates of missing values compared to other classes.

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

I would first eliminate predictors with more than 15% missing data, Hail, Sever, Seed Treatment, Shriveling, and Mold Growth. Then, impute missing data for important predictors using mode imputation (for categorical variables) or KNN imputation for more accurate results. Predictors like Precipitation, Temperature, Leaf-related attributes (Halo, Marg, Size, Shread, Malf), Plant Stand, Roots, and Stem. Lastly optionally eliminate rows from classes like 2-4-D-Injury and Cyst-Nematode if missingness is very high, or use class-specific imputation to handle missing data based on class.

By following this strategy of elimination for highly missing predictors and imputation for important predictors with moderate missing data, I can preserve the integrity of the Soybean dataset while minimizing the impact of missing values on model performance. Depending on my analysis, I can also consider handling class-specific missing data to ensure accurate classification.