DATA 624 - Homework 4

Richie Rivera

These questions come from the Applied Predictive Modeling book.

Question 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 4.3.3
```

```
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 ...
```

```
# ?Glass
```

```
library(GGally)
```

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

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

```
## Loading required package: ggplot2
```

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

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

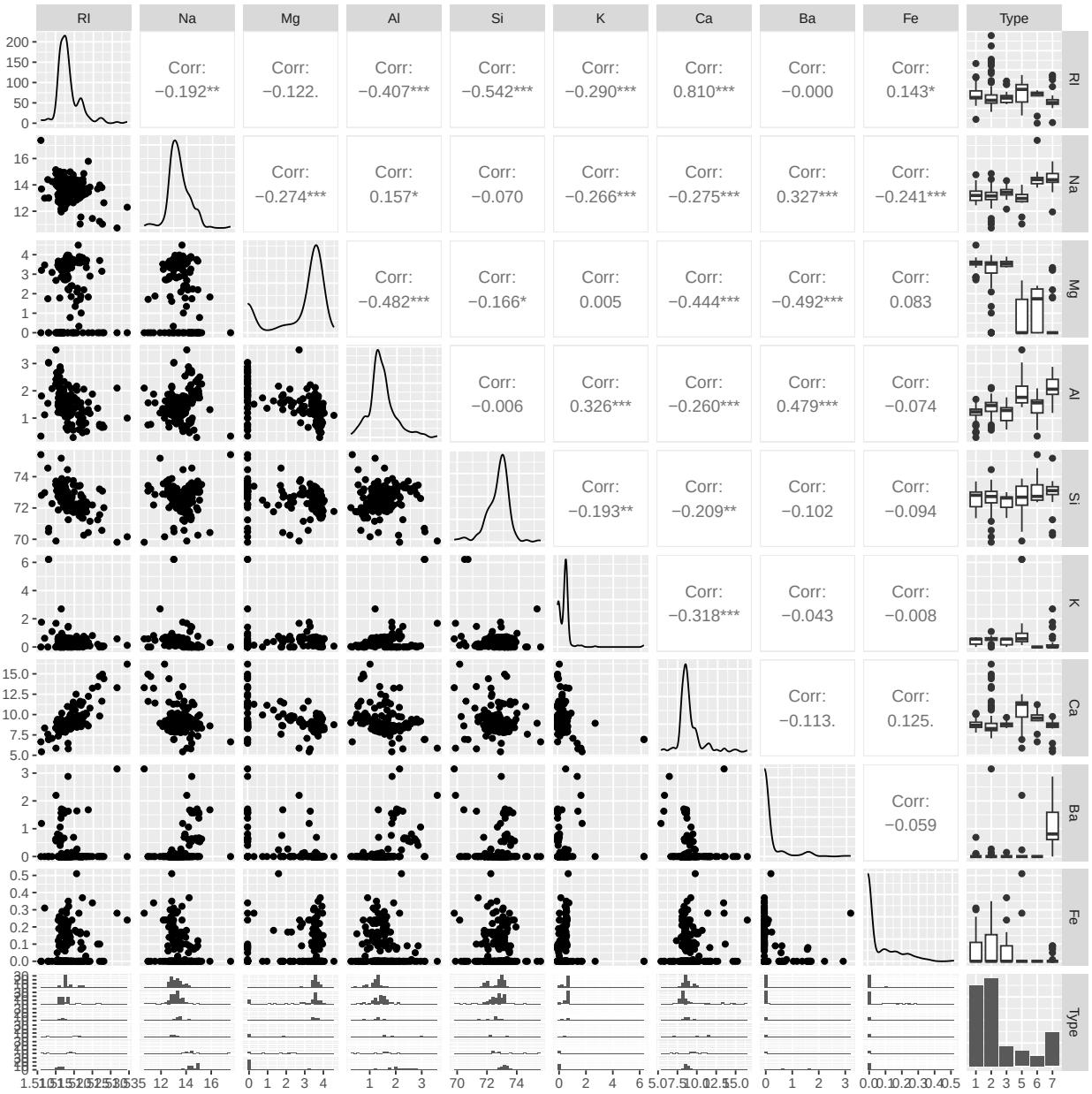
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
Glass |>
  ggpairs()
```

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

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



Above I am using `ggpairs()` as it will return a lot of useful visualizations for our data. Along the diagonal, we can see the distribution of each feature across this dataset. Each histogram is relative to the maximum and minimum values present in the provided data.

By looking into here, I see:

1. `RI` - is right skewed
2. `Na` - is fairly normally distributed
3. `Mg` - seems to have a good portion of low values and then many more values that are close to the upper bound
4. `Al` - is a little normal but right skewed
5. `Si` - the main component in glass. This seems fairly normal
6. `K` - is mostly distributed across very low values. This one seems to have a few outliers which we can see in the charts below it.
7. `Ca` - is right skewed and fairly normal

8. `Ba` - has almsot all of it's values at or very near to 0 with a few observations that have higher compositions.
9. `Fe` - values are all very low (less than 0.5) but they are more varied in their distribution than `Ba`.
10. `Type` - Because this is a categorical value, it's diagonal component shows a bar chart of frequency. Using that we see that types 1, 2, and 7 are the most common entries.

Moving towards the top right of the diagonal, we can see a series of correlation scores. This shows how well each feature is correlated with others. With that I see:

- `RI` and `Al` are fairly negatively correlated with a -40.7% correlation coefficient.
- `RI` and `Si` are fairly negatively correlated with a -54.2% correlation coefficient.
- `RI` and `Ca` are highly correlated with a 81% correlation coefficient.
- `Mg` and `Al` are fairly negatively correlated with a -48.2% correlation coefficient.
- `Mg` and `Ca` are fairly negatively correlated with a -44.4% correlation coefficient.
- `Mg` and `Ba` are fairly negatively correlated with a -49.2% correlation coefficient.
- The rest of the pairs have poor correlations.

The last thing I would like to take a look at are the `facet_wrapped()` histograms at the bottom. That shows the distributions of each element across each category in `Type`. Using that a few notable takeaways are:

- The distribution of `Na` is different for type 7 than the rest. Although types 1-6 are similar.
- The distribution of `Mg` is very right skewed for types 1, 2, and 3. For type 7, they all seems to be highly concentrated at 0.
- The vast majority of non-zero `Fe` values are concentrated in types 1 and 2.
- The vast majority of non-zero `Ba` values are concentrated in type 7.

(b) Do there appear to be any outliers in the data? Are any predictors skewed? The skew is mentioned in the above answer for question (a). Regarding outliers, there seems to be outliers in:

- `K` has value(s) which are much higher than the rest
- `Fe` has value(s) which are much higher than the rest

(c) Are there any relevant transformations of one or more predictors that might improve the classification model? Firstly I would attempt to correct the skew. I woud attempt to do so by using a box-cox transformation. By getting the guerro optimized lambda, I can refer to the lookup here to find the appropriate function for this transformation.

Next I would handle outliser, likely by removing the high outliers for `K` and `Ba`. I've chosen those because these high outliers don't seem to have any predictive power for determining type. To contrast, `Mg` has a few very low outliers but they seem to correlate with Type 7 and their predictive power seems useful.

Question 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)
# ?Soybean
```

```r
library(tidyr)

Soybean |>
  gather() |>
  ggplot(
    aes(value)
  ) +
  facet_wrap(
    ~ key,
    scales = "free",
    ncol = 3
  ) +
  geom_density() +
  geom_bar() +
  coord_flip()
```

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

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

```
## Warning: Groups with fewer than two data points have been dropped.
## Groups with fewer than two data points have been dropped.
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```


area.dam
canker.lesion
Class
crop.hist
date
ext.decay
fruit.pods
fruit.spots
fruiting.bodies
germ
hail
int.discolor
leaf.halo
leaf.malf
leaf.marg
leaf.mild
leaf.shread
leaf.size
leaves
lodging
mold.growth
mycelium
plant.growth
plant.stand
precip
roots
sclerotia
seed
seed.discolor
seed.size
seed.tmt
sever
shriveling
stem
stem.cankers
temp
value
count

With the definition of degenerates as that there is typically very few categories value with little variance (IE, 2 options and one option is strongly dominant), we can see that this is the case for `leaf.malf`, `mycelium`, `sclerotia`, and `mold.growth`.

```
Soybean |>
  select(- Class) |>
  gather() |>
  group_by(key) |>
  summarise(
    prop_NA = sum(is.na(value)) / n()
  ) |>
  arrange(desc(prop_NA))
```

(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?

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

```
## # A tibble: 35 x 2
##    key             prop_NA
##    <chr>             <dbl>
##  1 hail              0.177
##  2 lodging           0.177
##  3 seed.tmt          0.177
##  4 sever             0.177
##  5 germ              0.164
##  6 leaf.mild         0.158
##  7 fruit.spots       0.155
##  8 fruiting.bodies   0.155
##  9 seed.discolor     0.155
## 10 shriveling        0.155
## # i 25 more rows
```

the predictors that are most likely to be missing are `hail`, `lodging`, `seed.tmt`, and `server` with a missing rate of 17.7%.

```
library(reshape2)
```

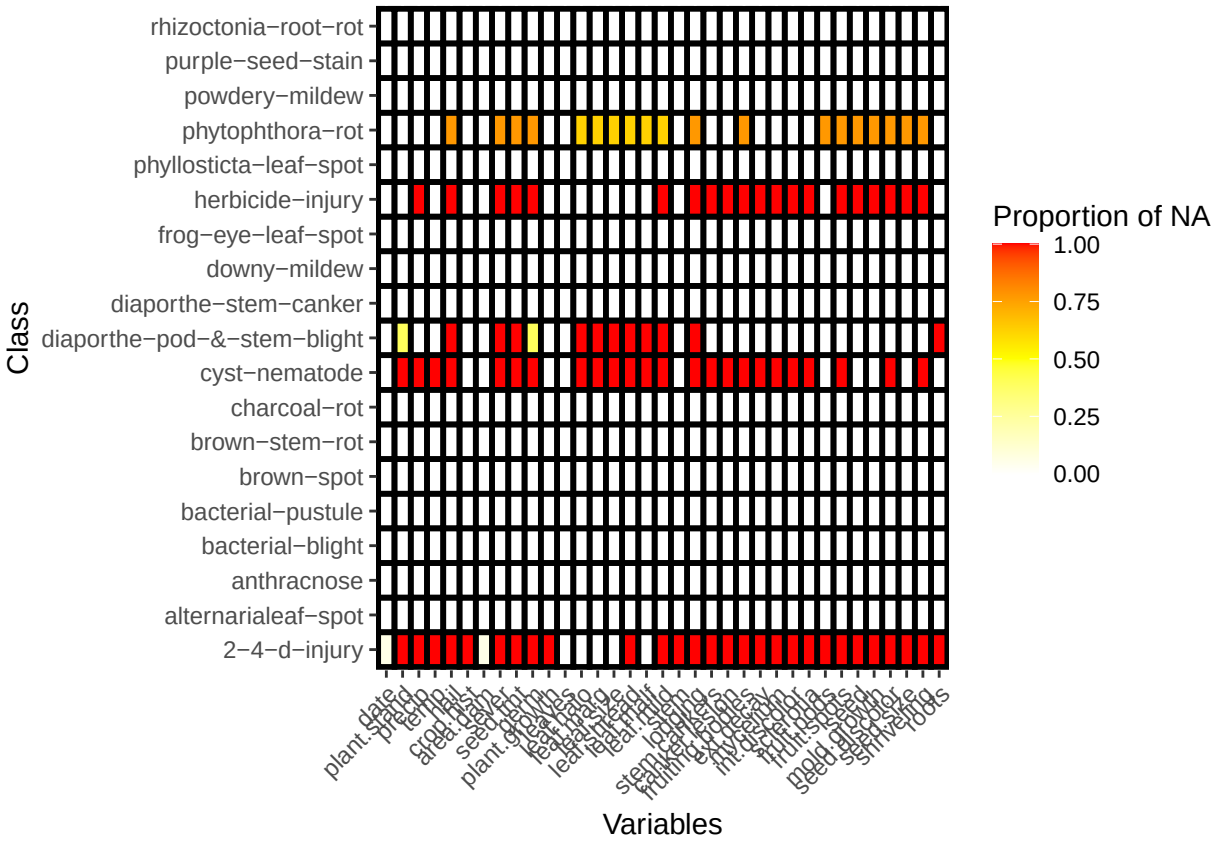
```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
na_by_class <- Soybean |>
  group_by(Class) |>
  summarise(across(everything(), ~ sum(is.na(.)) / n()), .groups = "drop")
```

```
na_by_class |>
  melt(id.vars = "Class") |>
  ggplot(aes(x = variable, y = Class, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(
    low = "white",
    mid = "yellow",
    high = "red",
    midpoint = 0.5, limits = c(0, 1)) +
  geom_tile(fill = NA, color = "black", size = 1) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
  ) +
  labs(x = "Variables", y = "Class", fill = "Proportion of NA")
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



Judging from the heatmap above, the `2-4-d-injury`, `cyst-nematode`, `diaporthe-pod-&-stem-blight`, and `herbicide-injury` have many predictors that are always missing while `phytophthora-rot` has some entries are missing but not all.

(c) Develop a strategy for handling missing data, either by eliminating predictors or imputation. It depends, for the classes that are missing this data it could be that the `<NA>` could encode other information that isn't specified in the `Soybean` dataset.

That being said, for this specific dataset and given our heatmap from before, I would recommend removing the `2-4-d-injury`, `cyst-nematode`, and `herbicide-injury` classes as there are too many predictors in the dataset with only `<NA>` values.

```
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.3.3
```

```
## Registered S3 method overwritten by 'tsibble':
##   method               from
##   as_tibble.grouped_df dplyr
```

```
## -- Attaching packages --------------------------------------------- fpp3 1.0.0 --
```

```
## v tibble      3.2.1     v feasts      0.3.2
## v lubridate   1.9.3     v fable       0.3.4
## v tsibble     1.1.5     v fabletools  0.4.2
## v tsibbledata 0.4.1
```

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

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

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

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

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

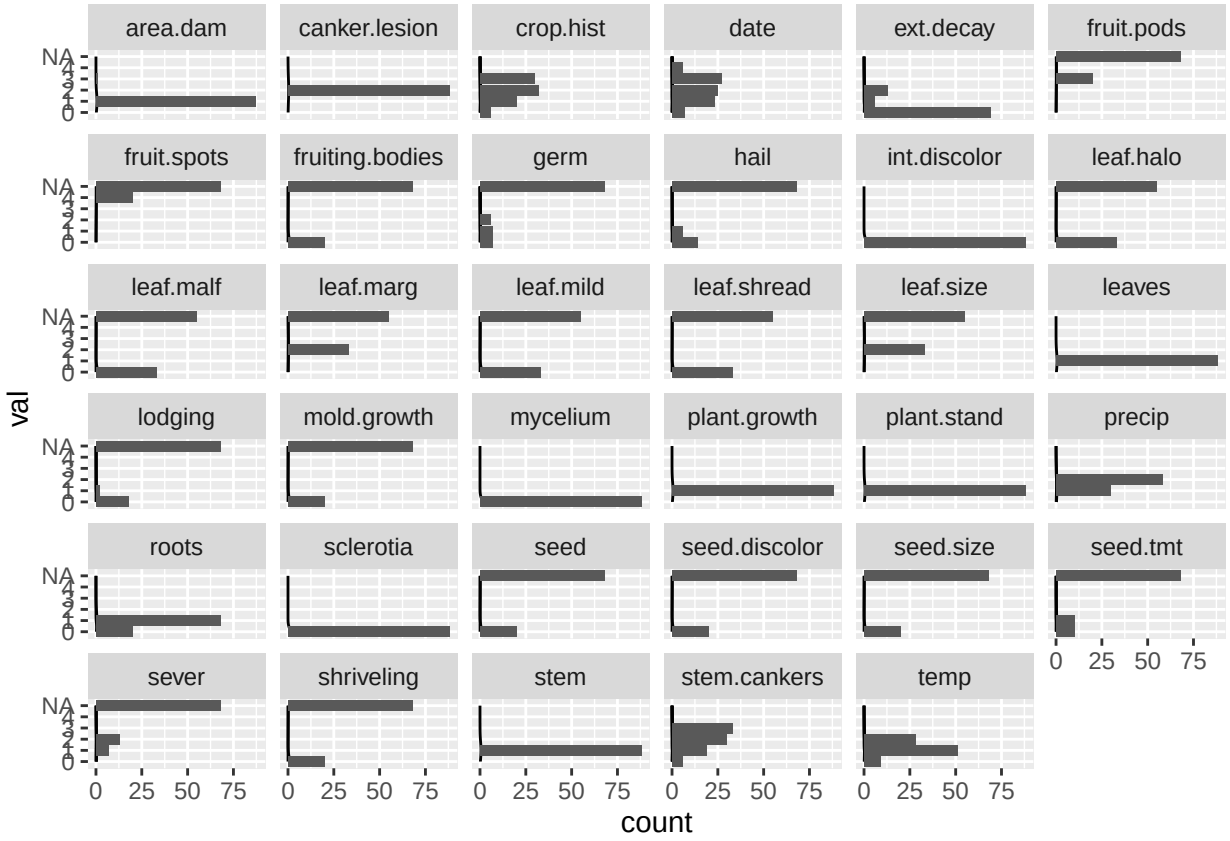
```
## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
Soybean |>
  filter(Class == "phytophthora-rot") |>
  pivot_longer(
    cols = -Class,
    names_to = "var",
    values_to = "val",
    values_ptypes = list(val = character())
  ) |>
  select(-Class) |>
```

```
  ggplot(
    aes(val)
  )  +
  facet_wrap(~var) +
  geom_density() +
  geom_bar() +
  coord_flip()
```

```
## Warning: Groups with fewer than two data points have been dropped.
```

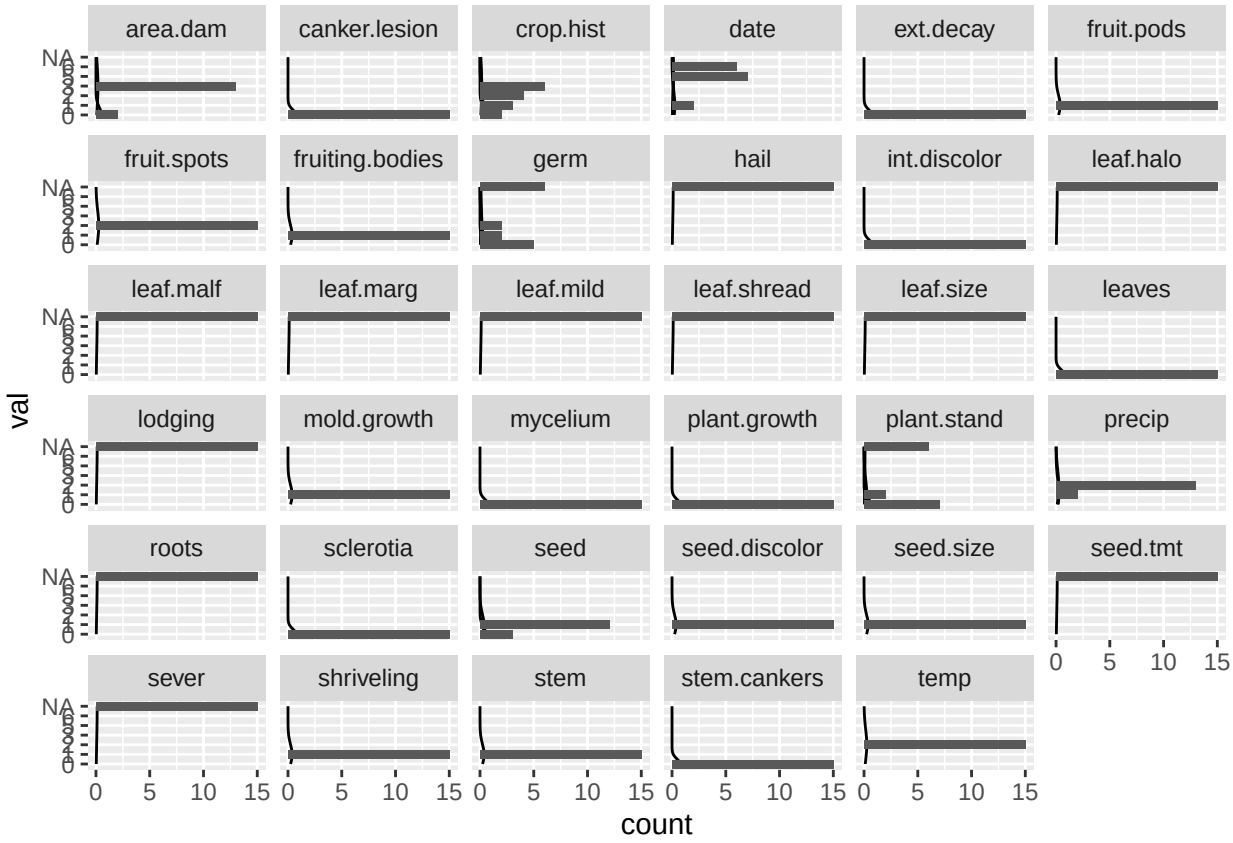
```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```



After seeing this chart, I would recommend to remove the `phytophthora-rot` class as well as the majority of values for this class are one value or `<NA>`.

```
Soybean |>
  filter(Class == "diaporthe-pod-&-stem-blight") |>
  pivot_longer(
    cols = -Class,
    names_to = "var",
    values_to = "val",
    values_ptypes = list(val = character())
  ) |>
  select(-Class) |>
```

```
ggplot(
  aes(val)
)  +
facet_wrap(~var) +
geom_density() +
geom_bar() +
coord_flip()
```



Looking at the plot for `diaporthe-pod-&-stem-blight`, I see that almost all of the entries where there are more values than just `<NA>` are very concentrated in one category. I would recommend removing this class as well for this reason.