

Data624 - Homework4

Amit Kapoor

2/28/2021

Contents

3.1	1
3.2	9

```
library(mlbench)
library(DataExplorer)
library(GGally)
library(psych)
library(caret)
library(summarytools)
library(naniar)
```

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.

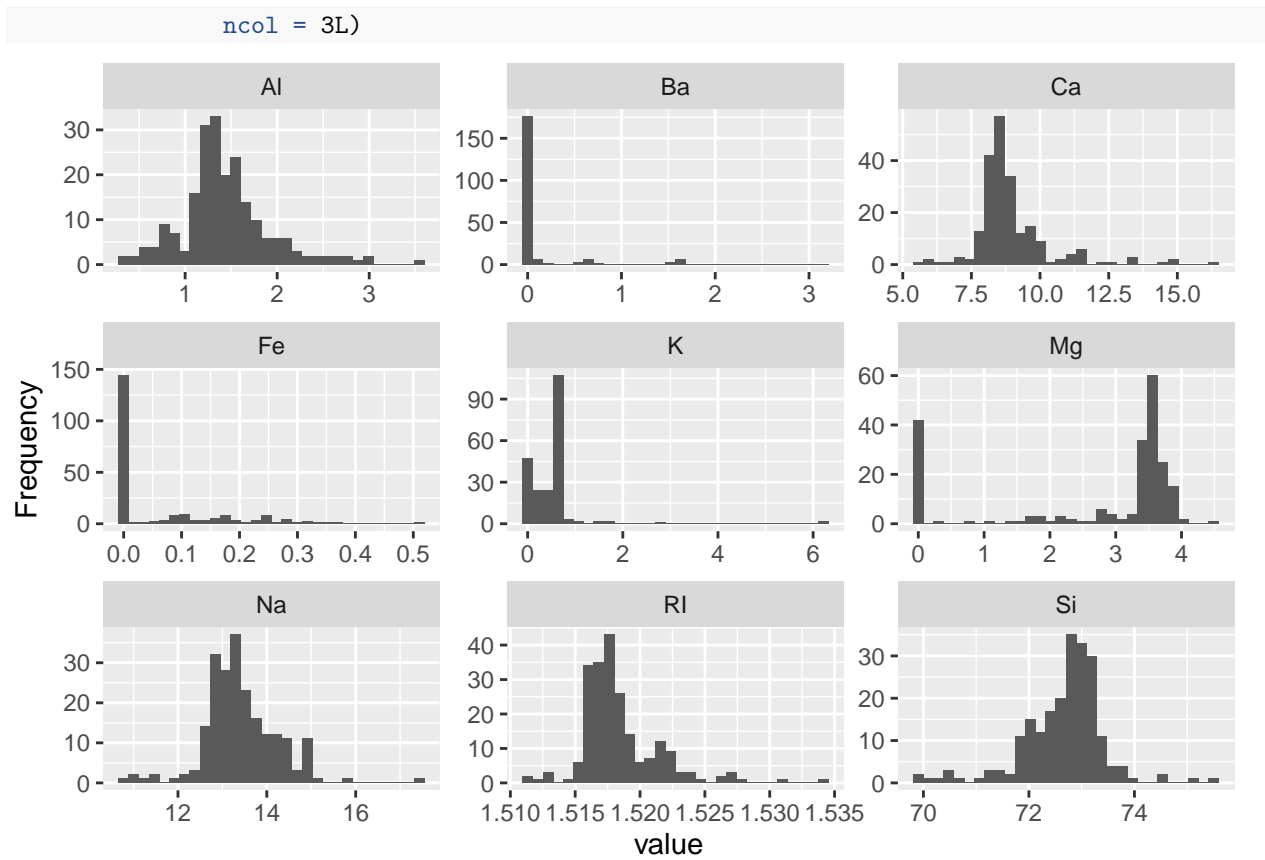
The data can be accessed via:

```
# load Glass data
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.

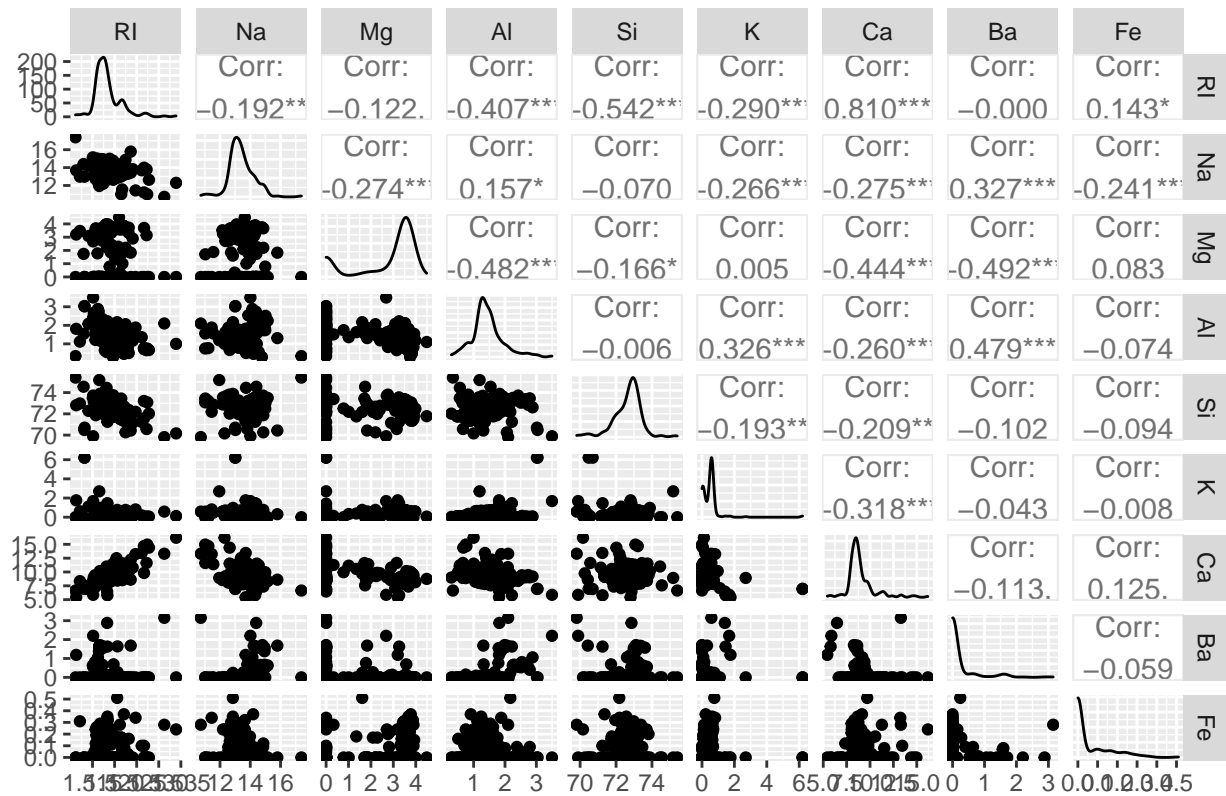
```
# predictors distribution
plot_histogram(Glass,
               geom_histogram_args = list(bins = 30L),
               nrow = 3L,
```



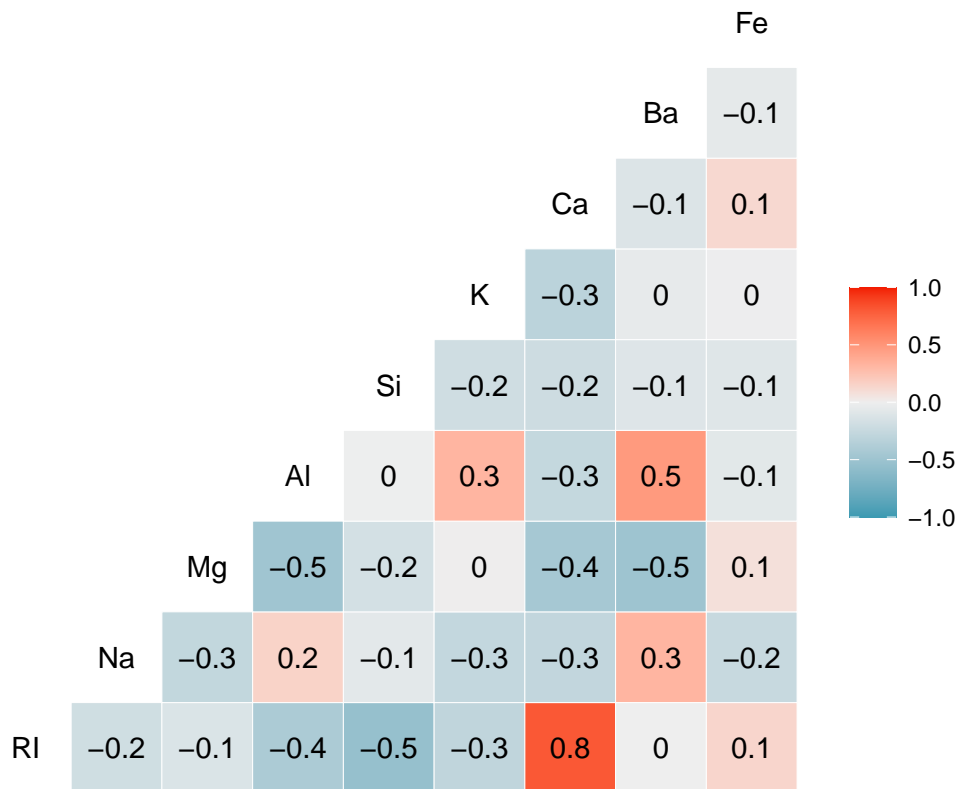
The plots above represent the distribution of predictors in the data. They could be used to compare the distributions against the normal distribution. Al, Na, Si appears nearly normal distributed with slight skewness while Ba, Ca, Fe, k and RI are right skewed.

```
# scatterplot matrix
Glass %>%
  dplyr::select(-Type) %>%
  ggpairs(title = "Pairwise scatter plots") %>%
  print(progress = F)
```

Paiwise scatter plots



```
# correlation
Glass %>%
  dplyr::select(-Type) %>%
  ggcorr(label = TRUE)
```



Above graphs depict the scatterplots and correlation among the predictors. It is evident that RI and Ca are strongly correlated. There are few other predictors having moderate correlation.

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

`describe(Glass)`

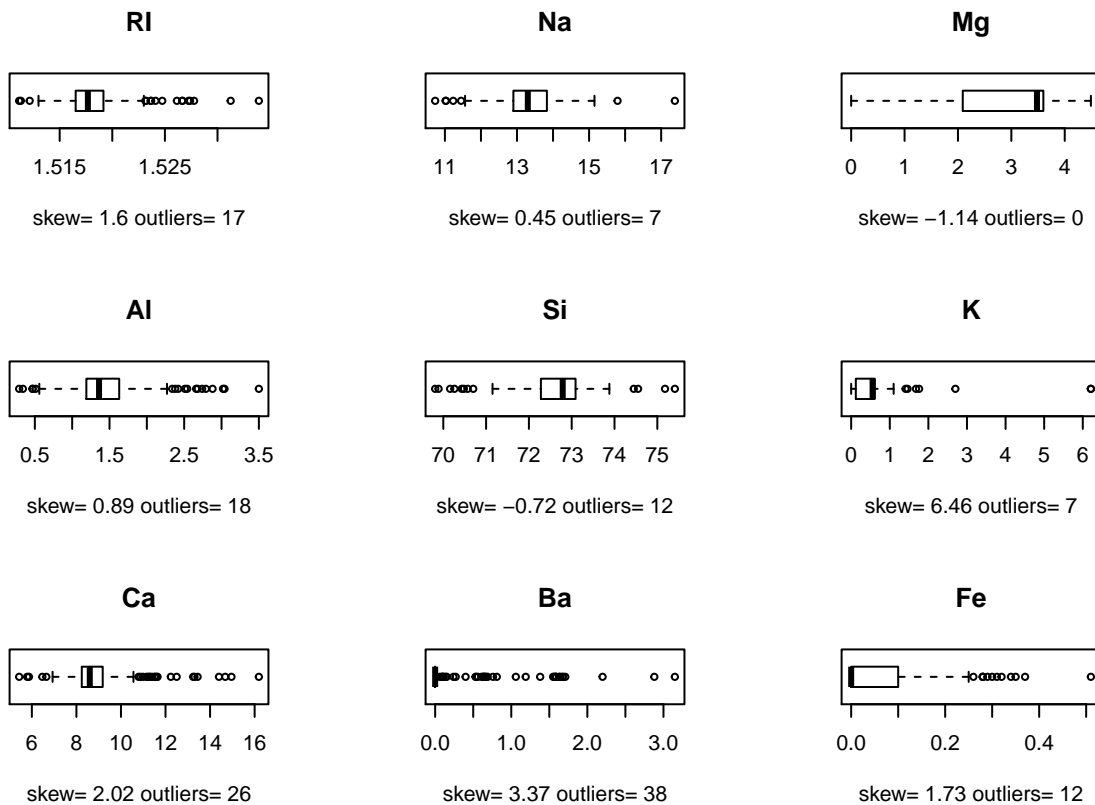
```
##      vars   n  mean   sd median trimmed  mad   min   max range  skew kurtosis
## RI         1 214  1.52 0.00   1.52   1.52 0.00  1.51  1.53  0.02  1.60    4.72
## Na         2 214 13.41 0.82  13.30  13.38 0.64 10.73 17.38  6.65  0.45    2.90
## Mg         3 214  2.68 1.44   3.48   2.87 0.30  0.00  4.49  4.49 -1.14   -0.45
## Al         4 214  1.44 0.50   1.36   1.41 0.31  0.29  3.50  3.21  0.89    1.94
## Si         5 214 72.65 0.77  72.79  72.71 0.57 69.81 75.41  5.60 -0.72    2.82
## K          6 214  0.50 0.65   0.56   0.43 0.17  0.00  6.21  6.21  6.46   52.87
## Ca         7 214  8.96 1.42   8.60   8.74 0.66  5.43 16.19 10.76  2.02    6.41
## Ba         8 214  0.18 0.50   0.00   0.03 0.00  0.00  3.15  3.15  3.37   12.08
## Fe         9 214  0.06 0.10   0.00   0.04 0.00  0.00  0.51  0.51  1.73    2.52
## Type*     10 214  2.54 1.71   2.00   2.31 1.48  1.00  6.00  5.00  1.04   -0.29
##          se
## RI         0.00
## Na         0.06
## Mg         0.10
## Al         0.03
## Si         0.05
## K          0.04
## Ca         0.10
## Ba         0.03
## Fe         0.01
## Type*     0.12
```

```

# function to get skewness and number of outliers for given var
label <- function(var) {
  return( paste("skew=", round(describe(var)$skew,2) , "outliers=" , length(boxplot(var, plot=FALSE)$outliers) ) )
}

par(mfrow=c(3,3))
# draw boxplot of predictors
for (i in 1:9){
  boxplot(
    Glass[i],
    color='green',
    horizontal = T,
    main = names(Glass)[i],
    xlab = label(Glass[i])
  )
}

```



With these boxplots, we can see the skewness measure and number of outliers for all predictors in the data. Ba and Ca have most of the outliers. Also Mg doesn't show up any outliers but its distribution seems left skewed.

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

As seen above, the variables in dataset are skewed so we will try the BoxCox transformation first to see if it improves them to an extent. Next we will try the transformation using BoxCox and PCA both if that makes any difference compared to the first one (with only BoxCox). The reason for using PCA is some variables showing correlations and we will see if PCA could improve that.

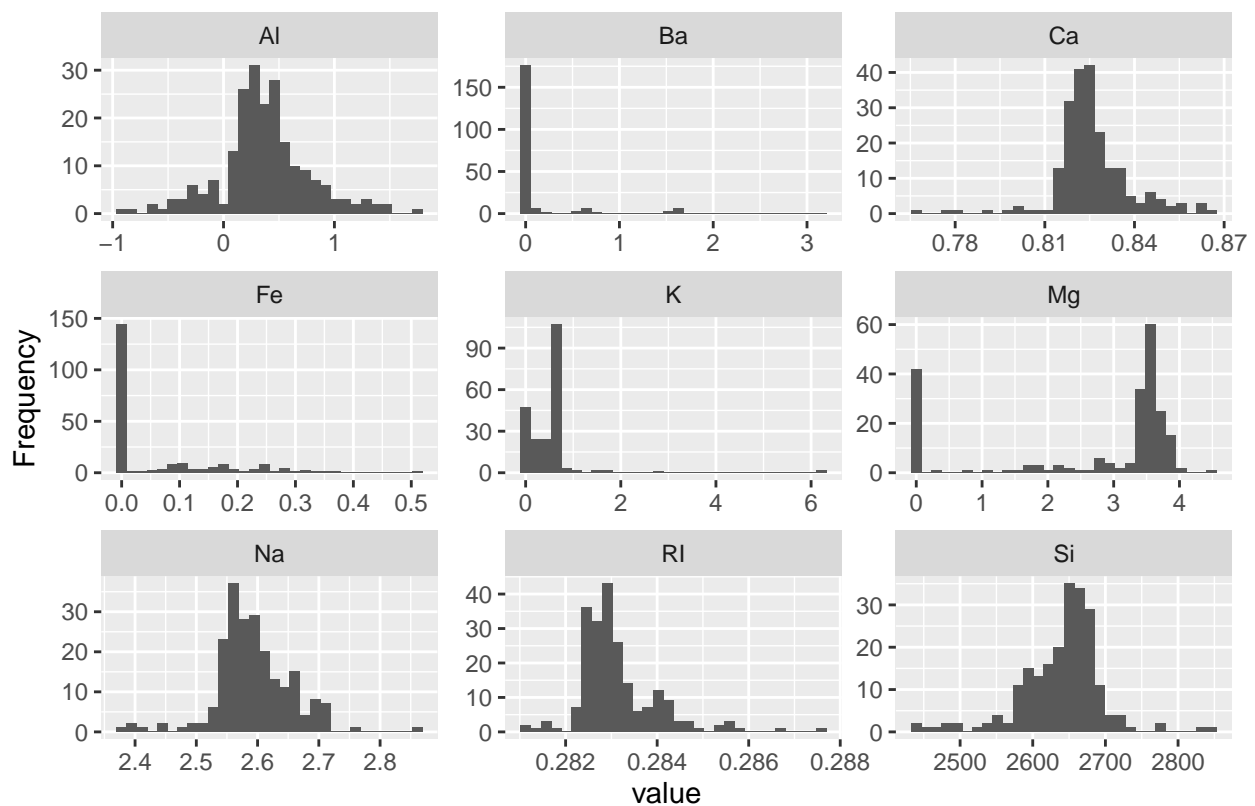
BoxCox Transformation

```
# preprocess using BoxCox
glass_boxcox_t <- preProcess(Glass, method = c("BoxCox"))
glass_boxcox_t
```

```
## Created from 214 samples and 6 variables
##
## Pre-processing:
## - Box-Cox transformation (5)
## - ignored (1)
##
## Lambda estimates for Box-Cox transformation:
## -2, -0.1, 0.5, 2, -1.1
```

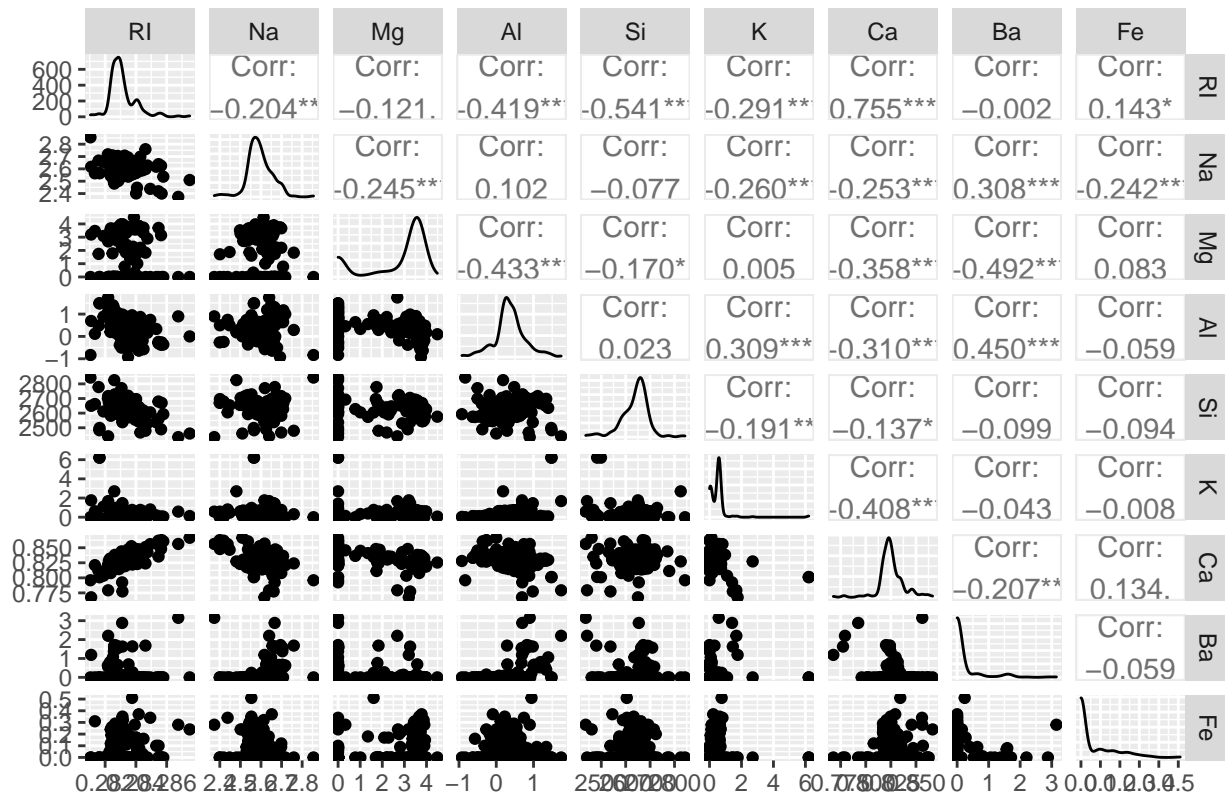
```
# predict using boxcox transformation
trans_boxcox <- predict(glass_boxcox_t, Glass)
```

```
# plot histogram
plot_histogram(trans_boxcox,
               geom_histogram_args = list(bins = 30L),
               nrow = 3L,
               ncol = 3L)
```



```
# scatterplot matrix
trans_boxcox %>%
  dplyr::select(-Type) %>%
  ggpairs(title = "Paiwise scatter plots") %>%
  print(progress = F)
```

Paiwise scatter plots



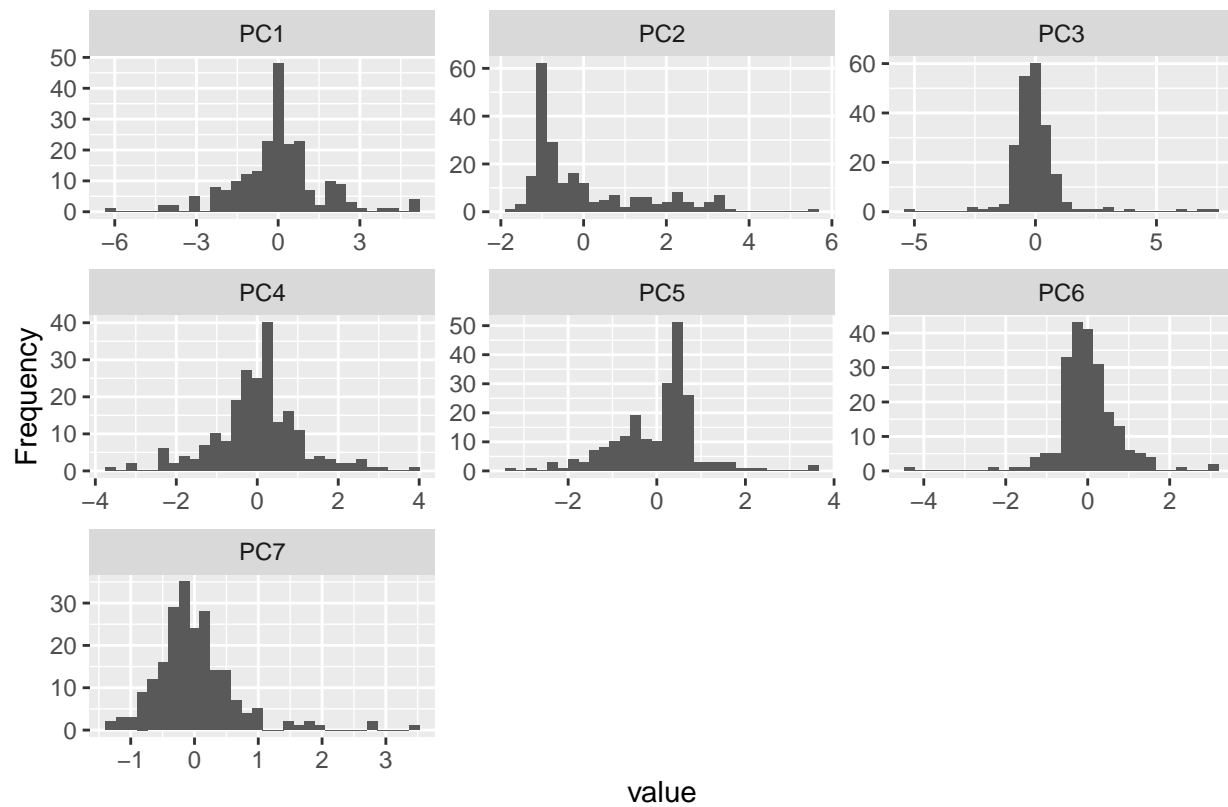
BoxCox Transformation and PCA

```
glass_bcpca_t <- preProcess(Glass, method = c("BoxCox", "pca"))
glass_bcpca_t

## Created from 214 samples and 10 variables
##
## Pre-processing:
##   - Box-Cox transformation (5)
##   - centered (9)
##   - ignored (1)
##   - principal component signal extraction (9)
##   - scaled (9)
##
## Lambda estimates for Box-Cox transformation:
## -2, -0.1, 0.5, 2, -1.1
## PCA needed 7 components to capture 95 percent of the variance

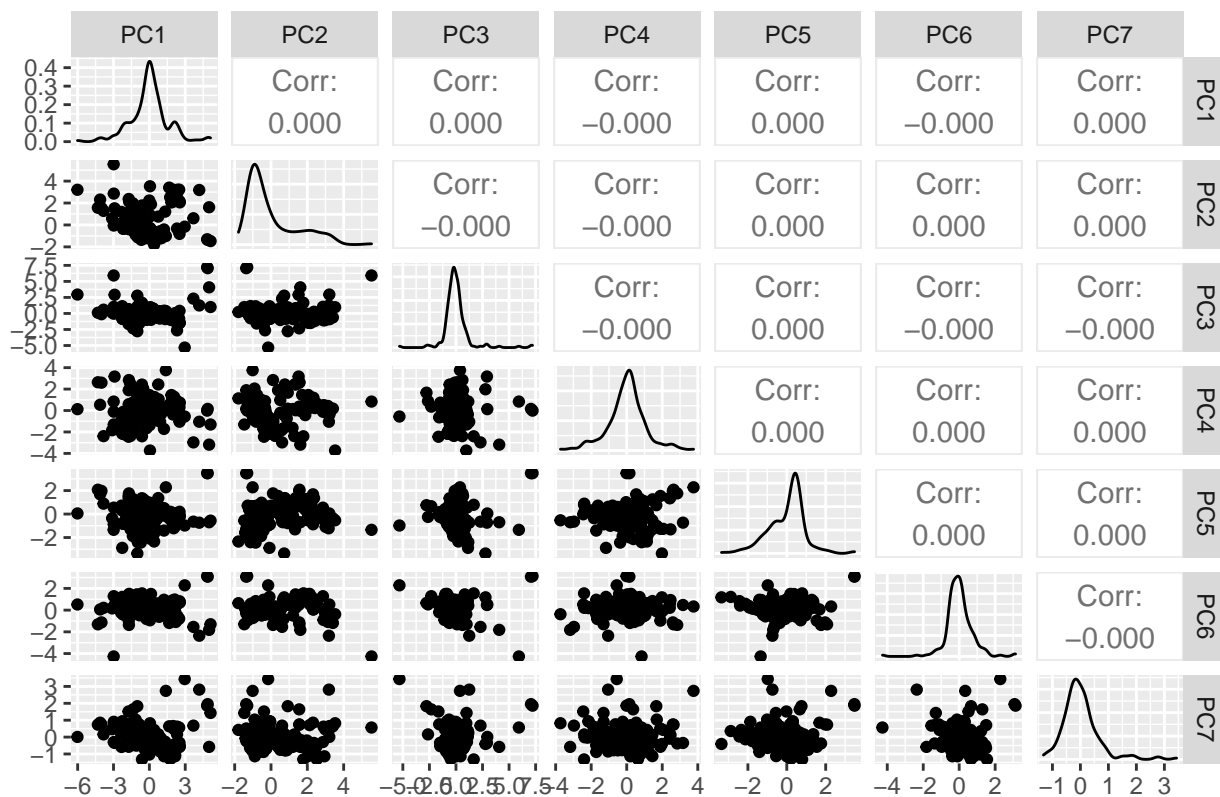
# predict with BoxCox and PCA transformation
trans_bcpca <- predict(glass_bcpca_t, Glass)

# plot histogram
plot_histogram(trans_bcpca,
               geom_histogram_args = list(bins = 30L),
               nrow = 3L,
               ncol = 3L)
```



```
# scatterplot matrix
trans_bcpca %>%
  dplyr::select(-Type) %>%
  ggpairs(title = "Paiwise scatter plots") %>%
  print(progress = F)
```


Paiwise scatter plots



We see BoxCox+PCA transformation makes the result better as compared to BoxCox only. BoxCox and PCA transformation makes the distribution a lot better and more towards normal distribution. Also it is evident that now it doesnt show any correlation.

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:

```
data(Soybean)
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 ...
```

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

```
# summary
dfSummary(Soybean, graph.col = F)
```

```
## Data Frame Summary
## Soybean
## Dimensions: 683 x 36
## Duplicates: 52
##
```

	No	Variable	Stats / Values	Freqs (% of Valid)	Valid	Missing
##	1	Class	1. 2-4-d-injury	16 (2.3%)	683	0
##		[factor]	2. alternarialeaf-spot	91 (13.3%)	(100.0%)	(0.0%)
##			3. anthracnose	44 (6.4%)		
##			4. bacterial-blight	20 (2.9%)		
##			5. bacterial-pustule	20 (2.9%)		
##			6. brown-spot	92 (13.5%)		
##			7. brown-stem-rot	44 (6.4%)		
##			8. charcoal-rot	20 (2.9%)		
##			9. cyst-nematode	14 (2.0%)		
##			10. diaporthe-pod-&-stem-blig	15 (2.2%)		
##			[9 others]	307 (44.9%)		
##	2	date	1. 0	26 (3.8%)	682	1
##		[factor]	2. 1	75 (11.0%)	(99.9%)	(0.1%)
##			3. 2	93 (13.6%)		

##		4. 3	118 (17.3%)		
##		5. 4	131 (19.2%)		
##		6. 5	149 (21.8%)		
##		7. 6	90 (13.2%)		
##					
## 3	plant.stand	1. 0	354 (54.7%)	647	36
##	[ordered, factor]	2. 1	293 (45.3%)	(94.7%)	(5.3%)
##					
## 4	precip	1. 0	74 (11.5%)	645	38
##	[ordered, factor]	2. 1	112 (17.4%)	(94.4%)	(5.6%)
##		3. 2	459 (71.2%)		
##					
## 5	temp	1. 0	80 (12.3%)	653	30
##	[ordered, factor]	2. 1	374 (57.3%)	(95.6%)	(4.4%)
##		3. 2	199 (30.5%)		
##					
## 6	hail	1. 0	435 (77.4%)	562	121
##	[factor]	2. 1	127 (22.6%)	(82.3%)	(17.7%)
##					
## 7	crop.hist	1. 0	65 (9.7%)	667	16
##	[factor]	2. 1	165 (24.7%)	(97.7%)	(2.3%)
##		3. 2	219 (32.8%)		
##		4. 3	218 (32.7%)		
##					
## 8	area.dam	1. 0	123 (18.0%)	682	1
##	[factor]	2. 1	227 (33.3%)	(99.9%)	(0.1%)
##		3. 2	145 (21.3%)		
##		4. 3	187 (27.4%)		
##					
## 9	sever	1. 0	195 (34.7%)	562	121
##	[factor]	2. 1	322 (57.3%)	(82.3%)	(17.7%)
##		3. 2	45 (8.0%)		
##					
## 10	seed.tmt	1. 0	305 (54.3%)	562	121
##	[factor]	2. 1	222 (39.5%)	(82.3%)	(17.7%)
##		3. 2	35 (6.2%)		
##					
## 11	germ	1. 0	165 (28.9%)	571	112
##	[ordered, factor]	2. 1	213 (37.3%)	(83.6%)	(16.4%)
##		3. 2	193 (33.8%)		
##					
## 12	plant.growth	1. 0	441 (66.1%)	667	16
##	[factor]	2. 1	226 (33.9%)	(97.7%)	(2.3%)
##					
## 13	leaves	1. 0	77 (11.3%)	683	0
##	[factor]	2. 1	606 (88.7%)	(100.0%)	(0.0%)
##					
## 14	leaf.halo	1. 0	221 (36.9%)	599	84
##	[factor]	2. 1	36 (6.0%)	(87.7%)	(12.3%)
##		3. 2	342 (57.1%)		
##					
## 15	leaf.marg	1. 0	357 (59.6%)	599	84
##	[factor]	2. 1	21 (3.5%)	(87.7%)	(12.3%)
##		3. 2	221 (36.9%)		

##						
## 16	leaf.size	1. 0	51 (8.5%)	599	84	
##	[ordered, factor]	2. 1	327 (54.6%)	(87.7%)	(12.3%)	
##		3. 2	221 (36.9%)			
##						
## 17	leaf.shread	1. 0	487 (83.5%)	583	100	
##	[factor]	2. 1	96 (16.5%)	(85.4%)	(14.6%)	
##						
## 18	leaf.malf	1. 0	554 (92.5%)	599	84	
##	[factor]	2. 1	45 (7.5%)	(87.7%)	(12.3%)	
##						
## 19	leaf.mild	1. 0	535 (93.0%)	575	108	
##	[factor]	2. 1	20 (3.5%)	(84.2%)	(15.8%)	
##		3. 2	20 (3.5%)			
##						
## 20	stem	1. 0	296 (44.4%)	667	16	
##	[factor]	2. 1	371 (55.6%)	(97.7%)	(2.3%)	
##						
## 21	lodging	1. 0	520 (92.5%)	562	121	
##	[factor]	2. 1	42 (7.5%)	(82.3%)	(17.7%)	
##						
## 22	stem.cankers	1. 0	379 (58.8%)	645	38	
##	[factor]	2. 1	39 (6.0%)	(94.4%)	(5.6%)	
##		3. 2	36 (5.6%)			
##		4. 3	191 (29.6%)			
##						
## 23	canker.lesion	1. 0	320 (49.6%)	645	38	
##	[factor]	2. 1	83 (12.9%)	(94.4%)	(5.6%)	
##		3. 2	177 (27.4%)			
##		4. 3	65 (10.1%)			
##						
## 24	fruiting.bodies	1. 0	473 (82.0%)	577	106	
##	[factor]	2. 1	104 (18.0%)	(84.5%)	(15.5%)	
##						
## 25	ext.decay	1. 0	497 (77.1%)	645	38	
##	[factor]	2. 1	135 (20.9%)	(94.4%)	(5.6%)	
##		3. 2	13 (2.0%)			
##						
## 26	mycelium	1. 0	639 (99.1%)	645	38	
##	[factor]	2. 1	6 (0.9%)	(94.4%)	(5.6%)	
##						
## 27	int.discolor	1. 0	581 (90.1%)	645	38	
##	[factor]	2. 1	44 (6.8%)	(94.4%)	(5.6%)	
##		3. 2	20 (3.1%)			
##						
## 28	sclerotia	1. 0	625 (96.9%)	645	38	
##	[factor]	2. 1	20 (3.1%)	(94.4%)	(5.6%)	
##						
## 29	fruit.pods	1. 0	407 (67.9%)	599	84	
##	[factor]	2. 1	130 (21.7%)	(87.7%)	(12.3%)	
##		3. 2	14 (2.3%)			
##		4. 3	48 (8.0%)			
##						
## 30	fruit.spots	1. 0	345 (59.8%)	577	106	

```
##      [factor]          2. 1          75 (13.0%)          (84.5%)          (15.5%)
##                        3. 2          57 ( 9.9%)
##                        4. 4         100 (17.3%)
##
## 31  seed              1. 0          476 (80.5%)          591          92
##      [factor]          2. 1          115 (19.5%)          (86.5%)          (13.5%)
##
## 32  mold.growth       1. 0          524 (88.7%)          591          92
##      [factor]          2. 1           67 (11.3%)          (86.5%)          (13.5%)
##
## 33  seed.discolor     1. 0          513 (88.9%)          577          106
##      [factor]          2. 1           64 (11.1%)          (84.5%)          (15.5%)
##
## 34  seed.size         1. 0          532 (90.0%)          591          92
##      [factor]          2. 1           59 (10.0%)          (86.5%)          (13.5%)
##
## 35  shriveling        1. 0          539 (93.4%)          577          106
##      [factor]          2. 1           38 ( 6.6%)          (84.5%)          (15.5%)
##
## 36  roots             1. 0          551 (84.5%)          652          31
##      [factor]          2. 1           86 (13.2%)          (95.5%)          (4.5%)
##                        3. 2           15 ( 2.3%)
## -----
```

This table shows the frequency distribution of Soybean dataset. The data has 683 rows and 36 variables. There are 35 predictors in the data and 'Class' is the response variable. There are 19 classes for response variable 'Class'. All predictors are numeric categorical variables.

From link, A degenerate distribution (sometimes called a constant distribution) is a distribution of a degenerate random variable — a constant with probability of 1. In other words, a random variable X has a single possible value. In other words, A random variable, X , is degenerate if, for some a constant, c , $P(X = c) = 1$.

The `nearZeroVar` function could be used to find the degenerate variables here.

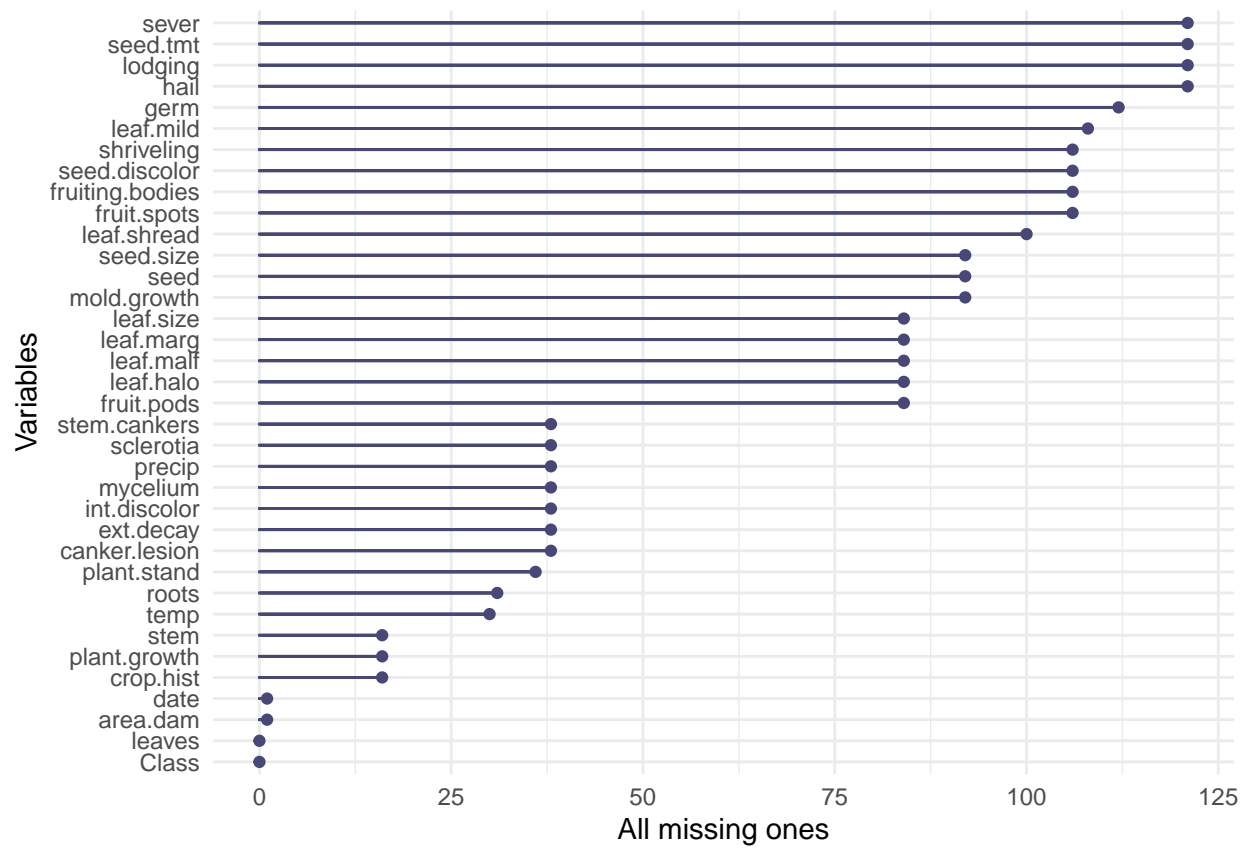
```
# find degenerate vars
nzvs <- nearZeroVar(Soybean)
names(Soybean)[nzvs]
```

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

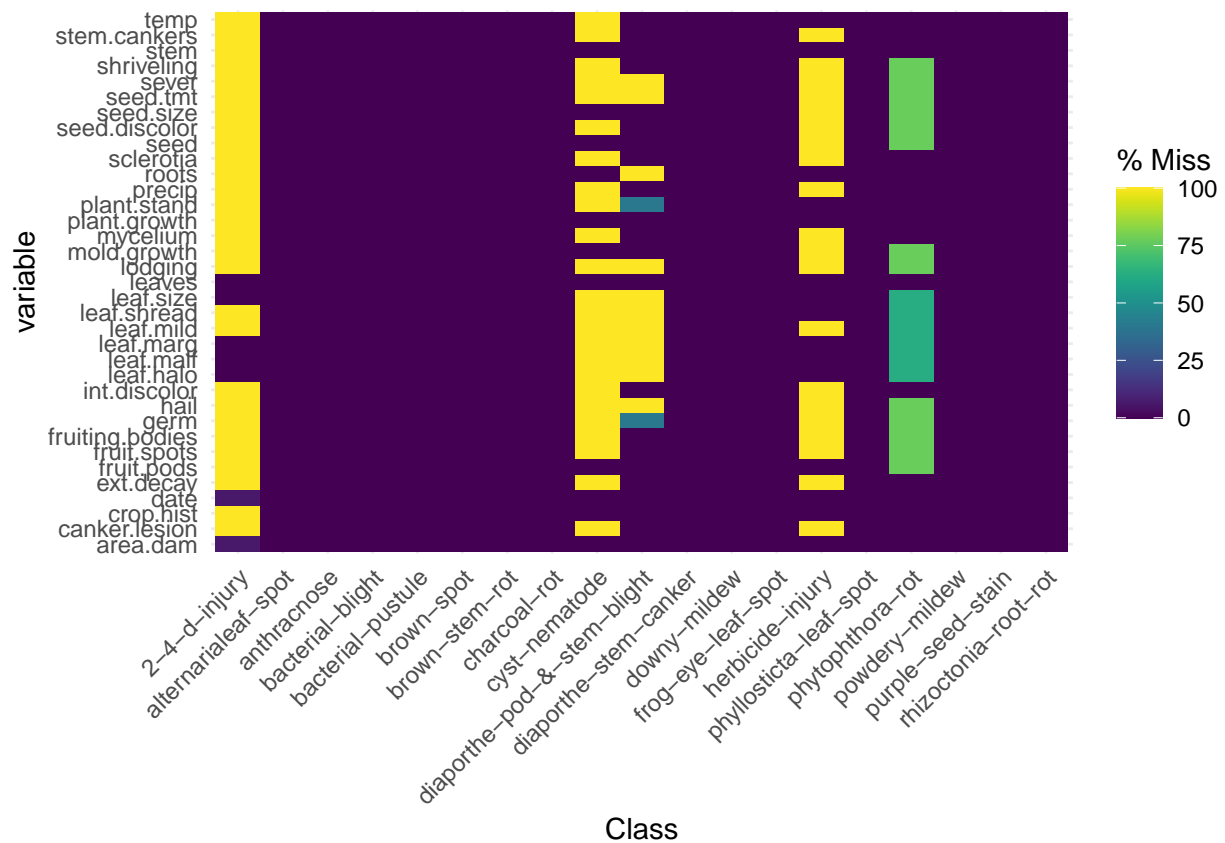
There are three variables `leaf.mild`, `mycelium` and `sclerotia` that have a non zero variance and it would be good to remove these variables from the model.

(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?

```
gg_miss_var(Soybean) + labs(y = "All missing ones")
```



```
gg_miss_fct(x=Soybean, fct=Class)
```



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