Data 624 HW 3

David Quarshie

9/9/2019

```
library('corrplot')
```

```
## corrplot 0.84 loaded
```

```
library('DataExplorer')
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from 
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
library('car')
```

```
## Loading required package: carData
```

```
library('caret')
```

```
## Loading required package: lattice
```

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

```
library('dplyr')
```

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

```
## The following object is masked from 'package:car':
## 
##     recode
```

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

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

```
library('tidyr')
library('mice')
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car 
##   influence.merMod                car 
##   dfbeta.influence.merMod         car 
##   dfbetas.influence.merMod        car
```

```
## 
## Attaching package: 'mice'
```

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

```
##     complete
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```r
library('VIM')
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## Loading required package: data.table
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
##
##           Please use the package to use the new (and old) GUI.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM/issues
```

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

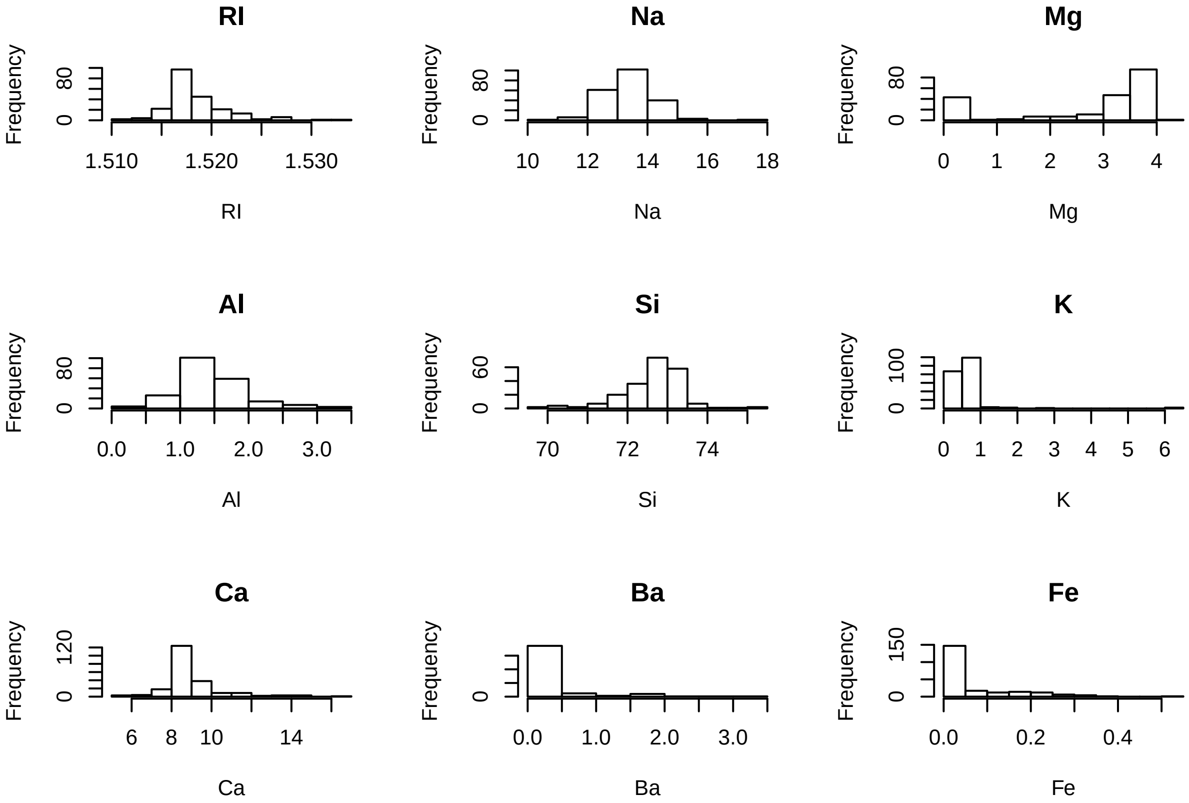
```r
library('mlbench')
data("Glass")
```

3.1

Part a & b : Explore Variables and Identify Outliers / Skewness

Histogram of predictor variables:

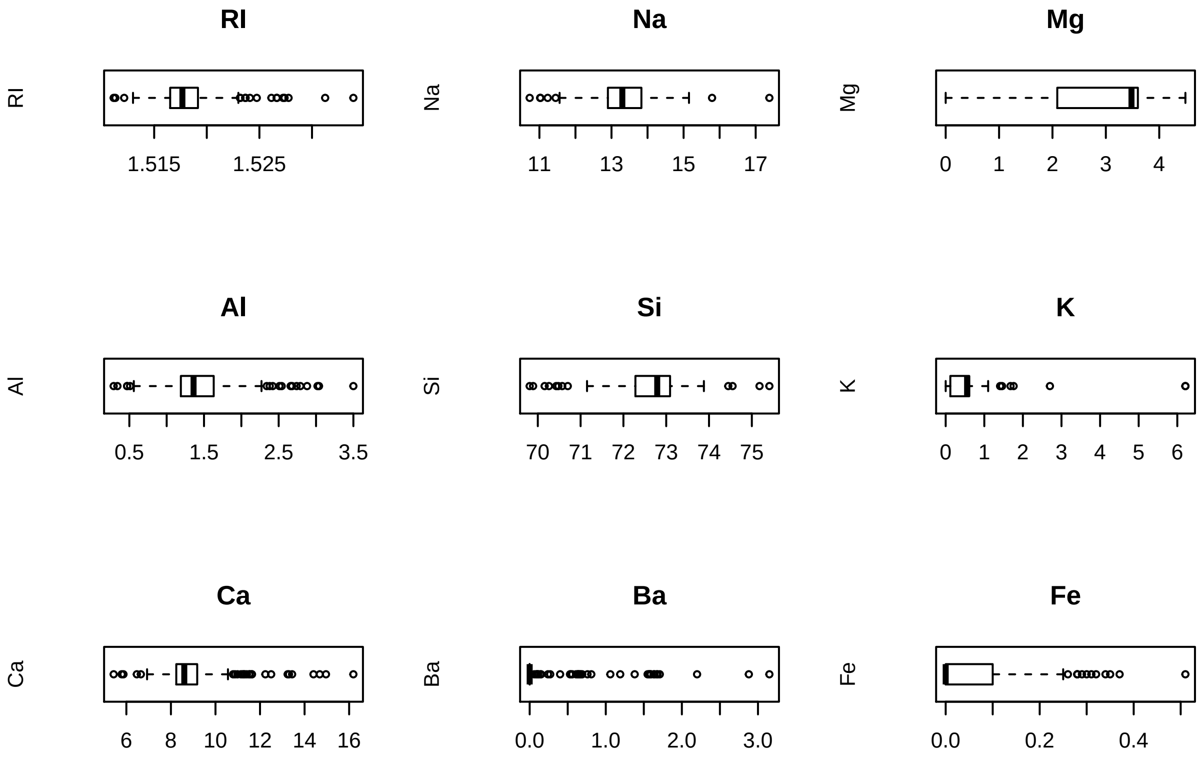
```r
par(mfrow = c(3, 3))
for (i in 1:ncol(Glass[,1:9])) {
  hist(Glass[ ,i], xlab = names(Glass[i]), main = names(Glass[i]))
}
```



Ploting histograms for each variable allows to examine their distributions. It looks like Ri, Na, Al, and Si have relatively normal distributions, while the others are skewed left or right. Ca, K, Ba, and Fe are right skewed while Mg is left skewed.

Boxplot of predictor variables:

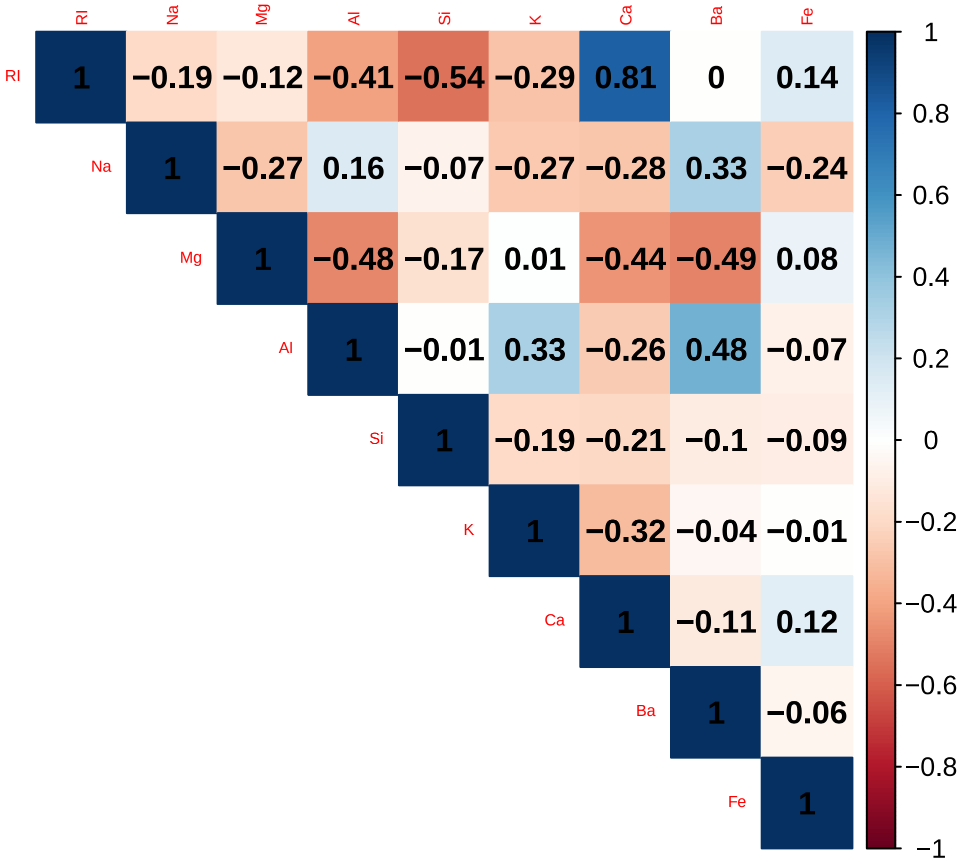
```r
par(mfrow = c(3, 3))
for (i in 1:ncol(Glass[,1:9])) {
  boxplot(Glass[,1:9][ ,i], ylab = names(Glass[,1:9][i]), horizontal=TRUE,
          main = paste(names(Glass[i])))
}
```



The boxplots for each variable verify that Ri, Na, Al, and Si are normally distributed but they also show that mostly every variable contains outliers. Mg seems to be the only variable with no outliers.

Correlation plot of predictor variables:

```r
# Feature Correlation plot
corrplot(cor(Glass[,1:9]),method='color',tl.cex=.5, type = "upper",addCoef.col = "black")
```



In regards to the relationships between the variables, we can use a correlation plot to view which variable are closely correlatied and which are not. The plot shows that Ri and Ca have high positive relationship with a value of 0.81. Ri also has the highest negative relationship, show a value of -0.54 with Si. Most of the variables seem to have little to no correlation to each other.

Part c: Variable Transformation: Box-Cox

Given that we know that the variables skew certain ways and contain outliers we will want to do data transformations to produce a usable prediciton model. The Box-Cox transformation is used to see how we should transform each variable so that it can be normalized to be used in a model. Using the powerTransform function each variable will be given a power that it should be raised to in order to be normalized.

```
summary(powerTransform(Glass[,1:9], family="bcnPower"))$result[,1:2]
```

```
##    Est Power Rounded Pwr
## RI -3.000000           1
## Na  3.000000           1
## Mg  3.000000           3
## Al  1.634025           1
## Si  3.000000           1
## K  -1.346118          -1
## Ca -2.999746          -1
## Ba -1.750300          -2
## Fe -1.834052          -2
```

Our Box-Cox transformation shows that we shouldn't change Ri, Na, Al, and Si, showing them having a power of 1. Those were also the variables that had relatively normal distributions. The suggestions are to raise the other skewed variables to certain powers.

3.2

```
data("Soybean")
```

Part a: Degenerte Distributions

```
X <- nearZeroVar(Soybean[,2:36], names = TRUE, saveMetrics=T)
X
```

```
##                  freqRatio percentUnique zeroVar   nzv
## date              1.137405     1.0248902   FALSE FALSE
## plant.stand       1.208191     0.2928258   FALSE FALSE
## precip            4.098214     0.4392387   FALSE FALSE
## temp              1.879397     0.4392387   FALSE FALSE
## hail              3.425197     0.2928258   FALSE FALSE
## crop.hist         1.004587     0.5856515   FALSE FALSE
## area.dam          1.213904     0.5856515   FALSE FALSE
## sever             1.651282     0.4392387   FALSE FALSE
## seed.tmt          1.373874     0.4392387   FALSE FALSE
## germ              1.103627     0.4392387   FALSE FALSE
## plant.growth      1.951327     0.2928258   FALSE FALSE
## leaves            7.870130     0.2928258   FALSE FALSE
## leaf.halo         1.547511     0.4392387   FALSE FALSE
## leaf.marg         1.615385     0.4392387   FALSE FALSE
## leaf.size         1.479638     0.4392387   FALSE FALSE
## leaf.shread       5.072917     0.2928258   FALSE FALSE
## leaf.malf        12.311111     0.2928258   FALSE FALSE
## leaf.mild        26.750000     0.4392387   FALSE  TRUE
## stem              1.253378     0.2928258   FALSE FALSE
## lodging          12.380952     0.2928258   FALSE FALSE
## stem.cankers      1.984293     0.5856515   FALSE FALSE
## canker.lesion     1.807910     0.5856515   FALSE FALSE
## fruiting.bodies   4.548077     0.2928258   FALSE FALSE
## ext.decay         3.681481     0.4392387   FALSE FALSE
## mycelium        106.500000     0.2928258   FALSE  TRUE
## int.discolor     13.204545     0.4392387   FALSE FALSE
## sclerotia        31.250000     0.2928258   FALSE  TRUE
## fruit.pods        3.130769     0.5856515   FALSE FALSE
## fruit.spots       3.450000     0.5856515   FALSE FALSE
## seed              4.139130     0.2928258   FALSE FALSE
## mold.growth       7.820896     0.2928258   FALSE FALSE
## seed.discolor     8.015625     0.2928258   FALSE FALSE
## seed.size         9.016949     0.2928258   FALSE FALSE
## shriveling       14.184211     0.2928258   FALSE FALSE
## roots             6.406977     0.4392387   FALSE FALSE
```

A variable can classified as degenerate if it's values have zero variance (one value) or near zero variance (small amount of different values). Using the nearZeroVar function we can see which variables have zero or near zero variance.

```
subset(X, zeroVar == TRUE | nzv == TRUE)
```

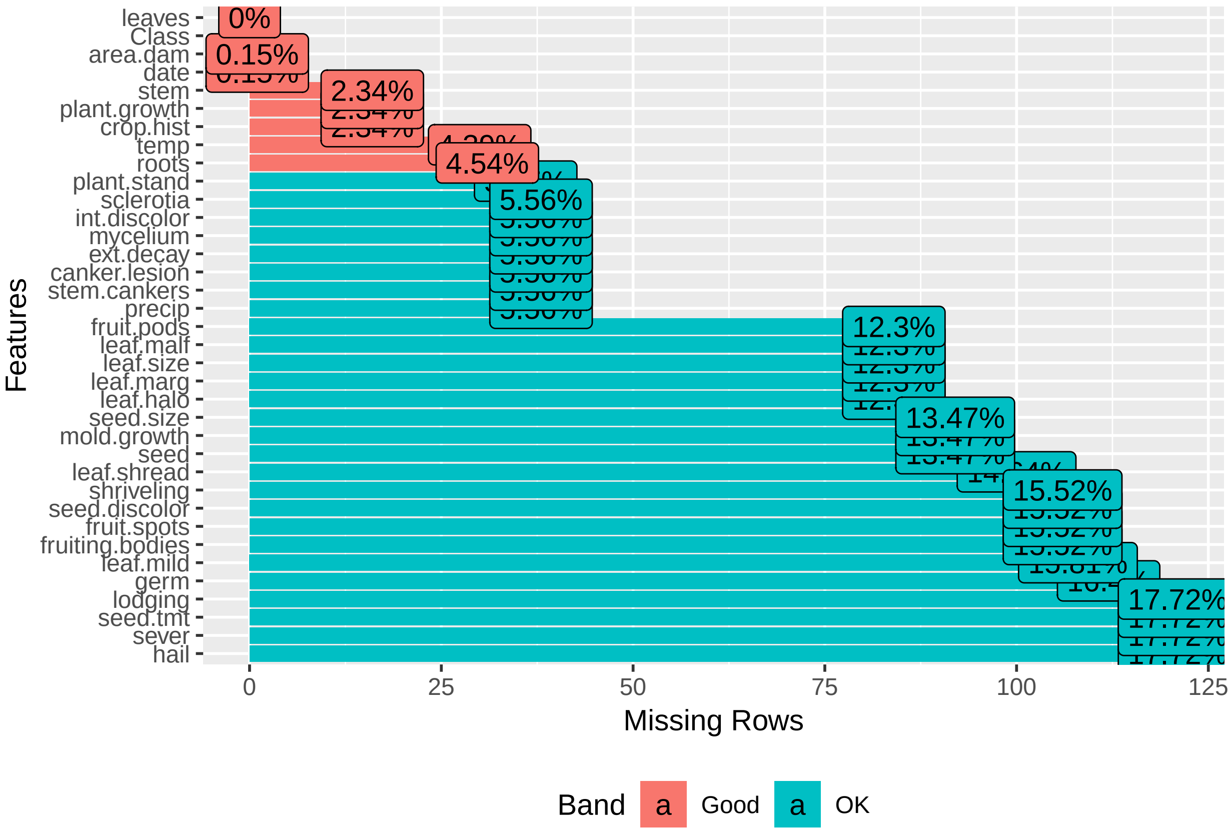
```
##           freqRatio percentUnique zeroVar  nzv
## leaf.mild     26.75     0.4392387   FALSE TRUE
## mycelium     106.50     0.2928258   FALSE TRUE
## sclerotia     31.25     0.2928258   FALSE TRUE
```

Running nearZeroVar shows that none of the variables have zero variance but leaf.mild, mycelium, and scelrotia all have near zero variance.

Part b: Pattern of Missing Data

We are told that 18% of the data is missing and by using the plot_missing function we can see which variables have missing values.

```
plot_missing(Soybean)
```



The plot_missing function shows that only leaves has no missing values and lodging, seed.tmt, server, and hall have the most missing values. Knowing this we can see if there's a pattern with missing data realted to the Class. By grouping by Class using the dplyr library we can see which have the most data missing.

```
Soybean %>%
  gather(Predictor, Value, -Class) %>%
  group_by(Class) %>%
  summarize(Missing = sum(is.na(Value))) %>%
  mutate(Missing = Missing / (nrow(Soybean) * 35)) %>%
  arrange(desc(Missing))
```

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

```
## # A tibble: 19 x 2
##    Class                       Missing
##    <fct>                         <dbl>
##  1 phytophthora-rot            0.0508
##  2 2-4-d-injury                0.0188
##  3 cyst-nematode               0.0141
##  4 diaporthe-pod-&-stem-blight 0.00740
##  5 herbicide-injury            0.00669
##  6 alternarialeaf-spot         0
##  7 anthracnose                 0
##  8 bacterial-blight            0
##  9 bacterial-pustule           0
## 10 brown-spot                  0
## 11 brown-stem-rot              0
## 12 charcoal-rot                0
## 13 diaporthe-stem-canker       0
## 14 downy-mildew                0
## 15 frog-eye-leaf-spot          0
## 16 phyllosticta-leaf-spot      0
## 17 powdery-mildew              0
## 18 purple-seed-stain           0
## 19 rhizoctonia-root-rot        0
```

Looking at missing values by Class, dplyr shows that the phytophthora-rot, 2-4-d-injury, cyst-nematode, diaporthe-pod&stem-blight, and herbicide-injury Classes have missing values.

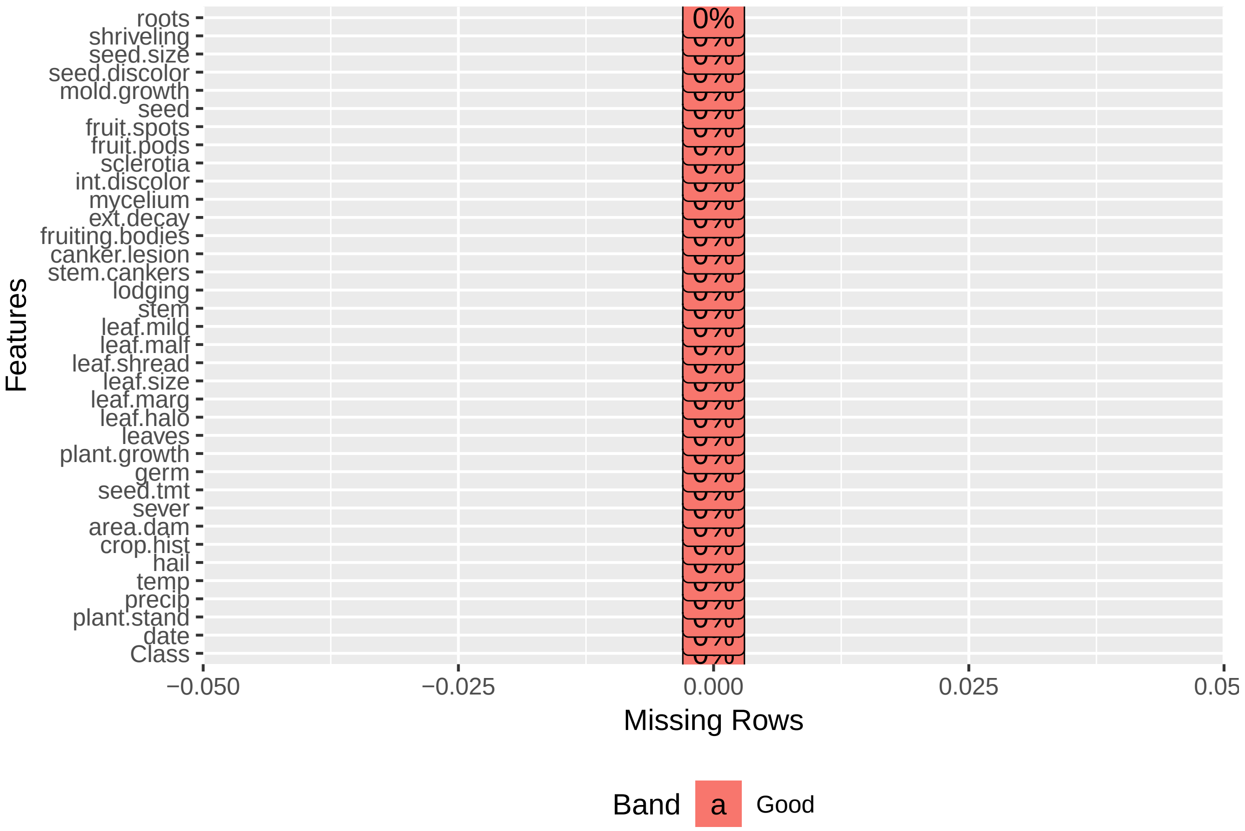
Part c: Handling Missing Data

Knowing that certain sections of the data contain missing data, we have to deal with them before making any models. Using the mice function R applies Predictive Mean Matching (PMM) to impute missing data. After running PMM on the data we can see that there is no more missing data

```
Soybean_imputed <- mice(Soybean, method="pmm", printFlag=F, seed=100)
```

```
## Warning: Number of logged events: 1662
```

```
Soybean_final <- complete(Soybean_imputed)
plot_missing(Soybean_final)
```


Features
roots
shriveling
seed.size
seed.discolor
mold.growth
seed
fruit.spots
fruit.pods
sclerotia
int.discolor
mycelium
ext.decay
fruiting.bodies
canker.lesion
stem.cankers
lodging
stem
leaf.mild
leaf.malf
leaf.shread
leaf.size
leaf.marg
leaf.halo
leaves
plant.growth
germ
seed.tmt
sever
area.dam
crop.hist
hail
temp
precip
plant.stand
date
Class
0%
−0.050
−0.025
0.000
0.025
0.05
Missing Rows
Band
a
Good