Searching for Similarity – Classification

Name: Ryan Donaldson

Date: 03/20/2023

Dataset

Please click here (https://www.kaggle.com/datasets/uciml/mushroom-classification) to access the dataset used in this project.

Train and Test

We will divide the data into train and test sets using a 80/20 split. We will also load the dplyr package to make column and row operations and mutations easier. We will also load the glmnet package to help build our logistic regression classification model. We will also load the tidymodels package so we can have a better output summary from the logistic regression model. We will load the class and gmodels packages for helping with the kNN regression. We will load the rpart packages to help with decision tree regression.

```
mushrooms <- read.csv("mushrooms.csv", na.strings="NA", header=TRUE)
if(!require("dplyr")) {
  install.packages("dplyr")
  library("dplyr", warn.conflicts=FALSE)
}
```

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

```
if(!require("glmnet")) {
  install.packages("glmnet")
  library("glmnet", warn.conflicts=FALSE)
}
```

```
## Loading required package: glmnet
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```
if(!require("tidymodels")) {
  install.packages("tidymodels")
  library("tidymodels", warn.conflicts=FALSE)
}
```

```
## Loading required package: tidymodels
```

```
## ── Attaching packages ────────────────────────────────────── tidymodels 1.0.0 ──
```

```
## ✔ broom        1.0.4     ✔ rsample      1.1.1
## ✔ dials        1.1.0     ✔ tibble       3.2.1
## ✔ ggplot2      3.4.1     ✔ tidyr        1.3.0
## ✔ infer        1.0.4     ✔ tune         1.0.1
## ✔ modeldata    1.1.0     ✔ workflows    1.1.3
## ✔ parsnip      1.0.4     ✔ workflowsets 1.0.0
## ✔ purrr        1.0.1     ✔ yardstick    1.1.0
## ✔ recipes      1.0.5
```

```
## ── Conflicts ───────────────────────────────────────── tidymodels_conflicts() ──
## ✖ purrr::discard()  masks scales::discard()
## ✖ tidyr::expand()   masks Matrix::expand()
## ✖ dplyr::filter()   masks stats::filter()
## ✖ dplyr::lag()      masks stats::lag()
## ✖ tidyr::pack()     masks Matrix::pack()
## ✖ recipes::step()   masks stats::step()
## ✖ tidyr::unpack()   masks Matrix::unpack()
## ✖ recipes::update() masks Matrix::update(), stats::update()
## • Search for functions across packages at https://www.tidymodels.org/find/
```

```
if(!require("class")) {
  install.packages("class")
  library("class", warn.conflicts=FALSE)
}
```

```
## Loading required package: class
```

```
if(!require("gmodels")) {
  install.packages("gmodels")
  library("gmodels", warn.conflicts=FALSE)
}
```

```
## Loading required package: gmodels
```

```
if(!require("rpart")) {
  install.packages("rpart")
  library("rpart", warn.conflicts=FALSE)
}
```

```
## Loading required package: rpart
```

```
##
## Attaching package: 'rpart'
```

```
## The following object is masked from 'package:dials':
##
##     prune
```

```
if(!require("rpart.plot")) {
  install.packages("rpart.plot")
  library("rpart.plot", warn.conflicts=FALSE)
}
```

```
## Loading required package: rpart.plot
```

```
set.seed(1234)
i <- sample(1:nrow(mushrooms), nrow(mushrooms)*0.80,
replace=FALSE)
train <- mushrooms[i,]
test <- mushrooms[-i,]
```

Statistical Data Exploration

Next, we will run 5 R functions for data exploration of the data set using the training data. First, let's run the str() function to get a look into the format of the data.

```
str(train)
```

```
## 'data.frame':	6499 obs. of  23 variables:
##  $ class                   : chr  "p" "e" "p" "e" ...
##  $ cap.shape               : chr  "k" "k" "k" "k" ...
##  $ cap.surface             : chr  "s" "f" "s" "f" ...
##  $ cap.color               : chr  "n" "w" "n" "w" ...
##  $ bruises                 : chr  "f" "f" "f" "f" ...
##  $ odor                    : chr  "f" "n" "s" "n" ...
##  $ gill.attachment         : chr  "f" "f" "f" "f" ...
##  $ gill.spacing            : chr  "c" "w" "c" "w" ...
##  $ gill.size               : chr  "n" "b" "n" "b" ...
##  $ gill.color              : chr  "b" "p" "b" "g" ...
##  $ stalk.shape             : chr  "t" "e" "t" "e" ...
##  $ stalk.root              : chr  "?" "?" "?" "?" ...
##  $ stalk.surface.above.ring: chr  "k" "k" "s" "s" ...
##  $ stalk.surface.below.ring: chr  "k" "k" "s" "k" ...
##  $ stalk.color.above.ring  : chr  "w" "w" "p" "w" ...
##  $ stalk.color.below.ring  : chr  "w" "w" "w" "w" ...
##  $ veil.type               : chr  "p" "p" "p" "p" ...
##  $ veil.color              : chr  "w" "w" "w" "w" ...
##  $ ring.number             : chr  "o" "t" "o" "t" ...
##  $ ring.type               : chr  "e" "p" "e" "p" ...
##  $ spore.print.color       : chr  "w" "w" "w" "w" ...
##  $ population              : chr  "v" "s" "v" "s" ...
##  $ habitat                 : chr  "p" "g" "d" "g" ...
```

Next, let’s gather an overall basic summary of each column of our training data.

```
summary(train)
```

```
##     class            cap.shape          cap.surface         cap.color
##  Length:6499        Length:6499        Length:6499        Length:6499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##    bruises              odor          gill.attachment    gill.spacing
##  Length:6499        Length:6499        Length:6499        Length:6499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##   gill.size          gill.color         stalk.shape         stalk.root
##  Length:6499        Length:6499        Length:6499        Length:6499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##  stalk.surface.above.ring stalk.surface.below.ring stalk.color.above.ring
##  Length:6499              Length:6499              Length:6499
##  Class :character         Class :character         Class :character
##  Mode  :character         Mode  :character         Mode  :character
##  stalk.color.below.ring  veil.type          veil.color
##  Length:6499            Length:6499        Length:6499
##  Class :character       Class :character   Class :character
##  Mode  :character       Mode  :character   Mode  :character
##  ring.number          ring.type          spore.print.color   population
##  Length:6499        Length:6499        Length:6499        Length:6499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##    habitat
##  Length:6499
##  Class :character
##  Mode  :character
```

So, first let's just look at the first few rows.

```
head(train)
```

	class <chr>	cap.shape <chr>	cap.surface <chr>	cap.color <chr>	bruises <chr>	o... <chr>	gill.attachment <chr>	gill.spacing <chr>	gi <c
7452	p	k	s	n	f	f	f	c	n
8016	e	k	f	w	f	n	f	w	b
7162	p	k	s	n	f	s	f	c	n
8086	e	k	f	w	f	n	f	w	b
7269	p	x	y	e	f	s	f	c	n
1004	p	x	s	w	t	p	f	c	n

6 rows | 1-10 of 24 columns

Now, let's look at the last few rows of our training data.

```
tail(train)
```

	class	cap.shape	cap.surface	cap.color	bruises	o...	gill.attachment	gill.spacing	gi
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<c
8074	e	k	s	n	f	n	a	c	b
6929	e	x	y	p	t	n	f	c	b
2716	e	x	y	n	t	n	f	c	b
7406	e	b	s	n	f	n	a	c	b
888	e	f	y	y	t	a	f	c	b
1556	p	f	s	n	t	p	f	c	n

6 rows | 1-10 of 24 columns

Let's also explore the column names within our data set in case we need to reference them later when making predictions.

```
names(train)
```

```
##  [1] "class"                    "cap.shape"
##  [3] "cap.surface"              "cap.color"
##  [5] "bruises"                  "odor"
##  [7] "gill.attachment"          "gill.spacing"
##  [9] "gill.size"                "gill.color"
## [11] "stalk.shape"              "stalk.root"
## [13] "stalk.surface.above.ring" "stalk.surface.below.ring"
## [15] "stalk.color.above.ring"   "stalk.color.below.ring"
## [17] "veil.type"                "veil.color"
## [19] "ring.number"              "ring.type"
## [21] "spore.print.color"        "population"
## [23] "habitat"
```

Data Cleaning

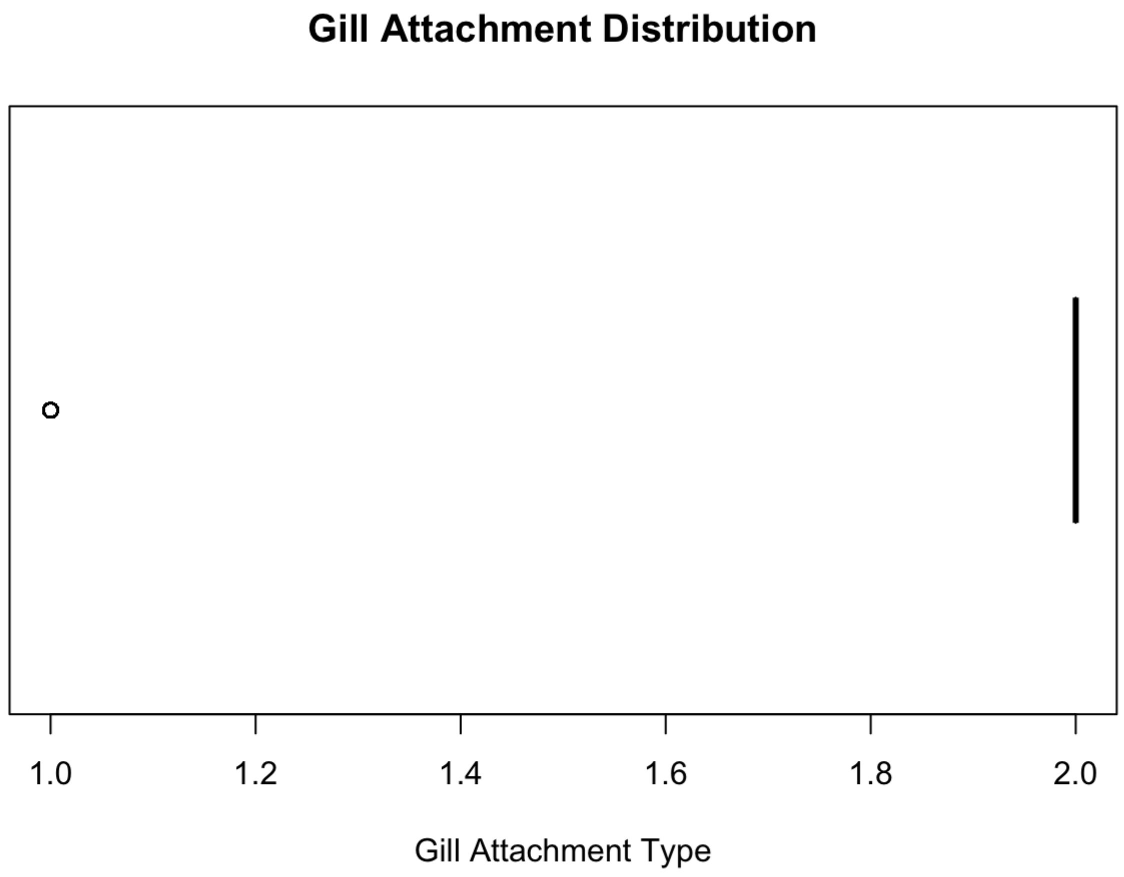
Before visualizing data, let's clean our dataset to make sure we can create the graphs and run the algorithms appropriately. We see as of right now we have characters representing the values of certain columns in the dataset. We will replace the various character values to a numeric factor except for the class column as this represents our target variable. Then, we will do an 80/20 split again.

```
new_train_df <- data.frame(sapply(train[2:23], function (new_train_df) as.numeric(as.fac
tor(new_train_df))))
train <- data.frame(new_train_df, class = train$class)
new_test_df <- data.frame(sapply(test[2:23], function (new_test_df) as.numeric(as.factor
(new_test_df))))
test <- data.frame(new_test_df, class = test$class)
```

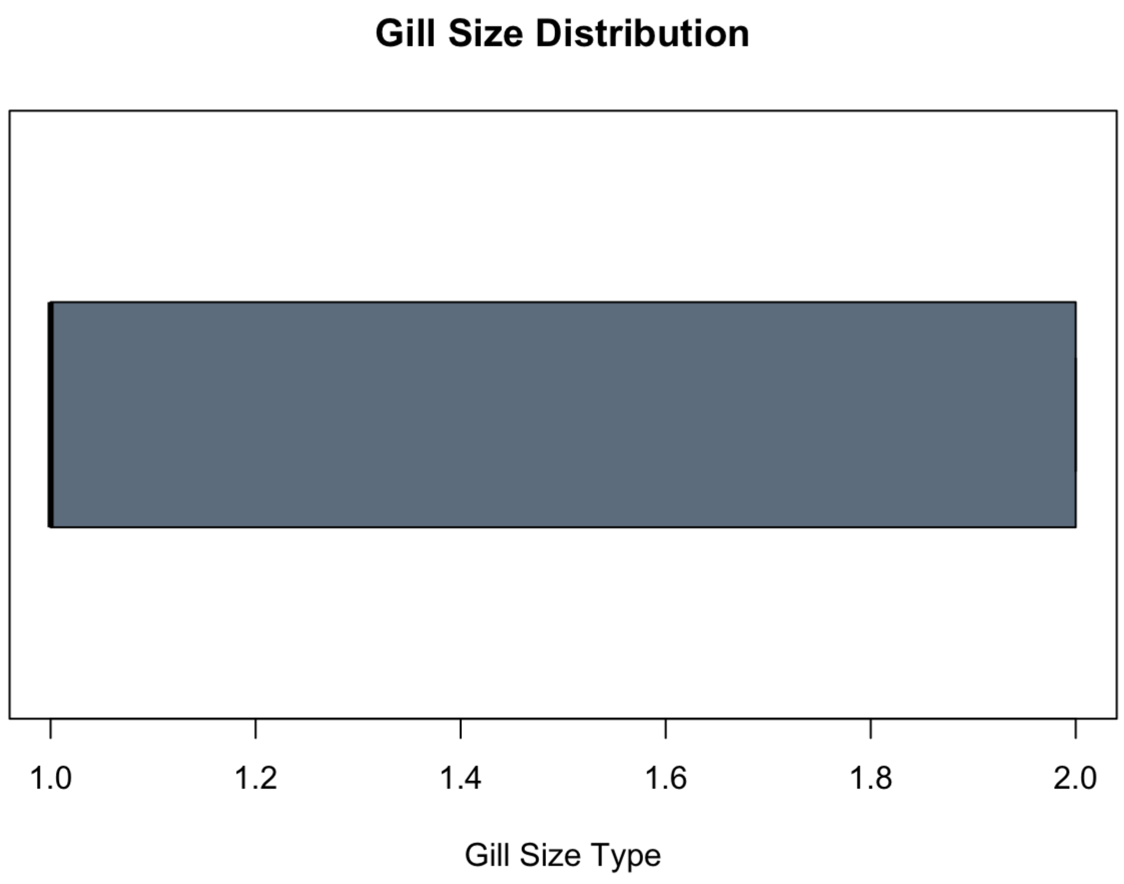
Graphical Data Exploration

Let’s create some informative graphs based on this training data, particuarly distributions of various columns since we’re working with such a large dataset. First, let’s get a sense of the distribution of the gill attachment, size, spacing, and color.

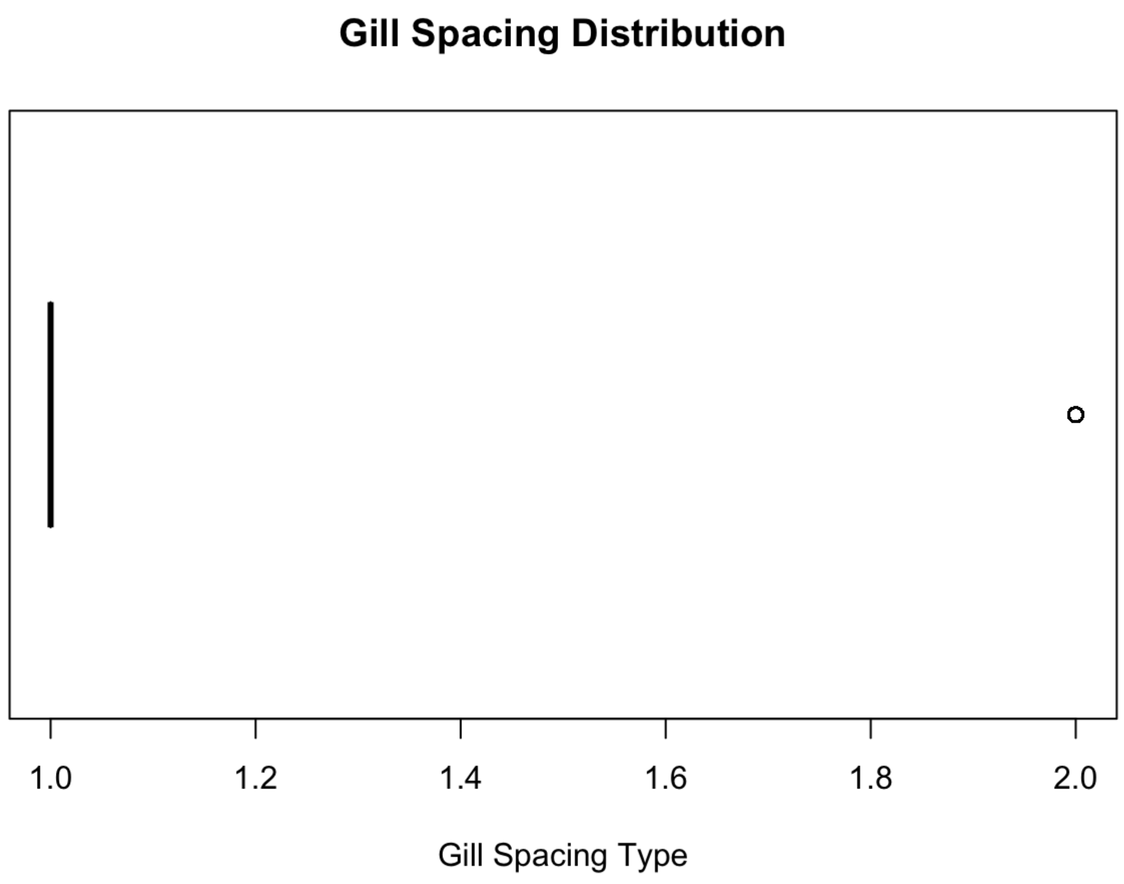
```
boxplot(train$gill.attachment, col="slategray", horizontal=TRUE, xlab="Gill Attachment T
ype", main="Gill Attachment Distribution")
```



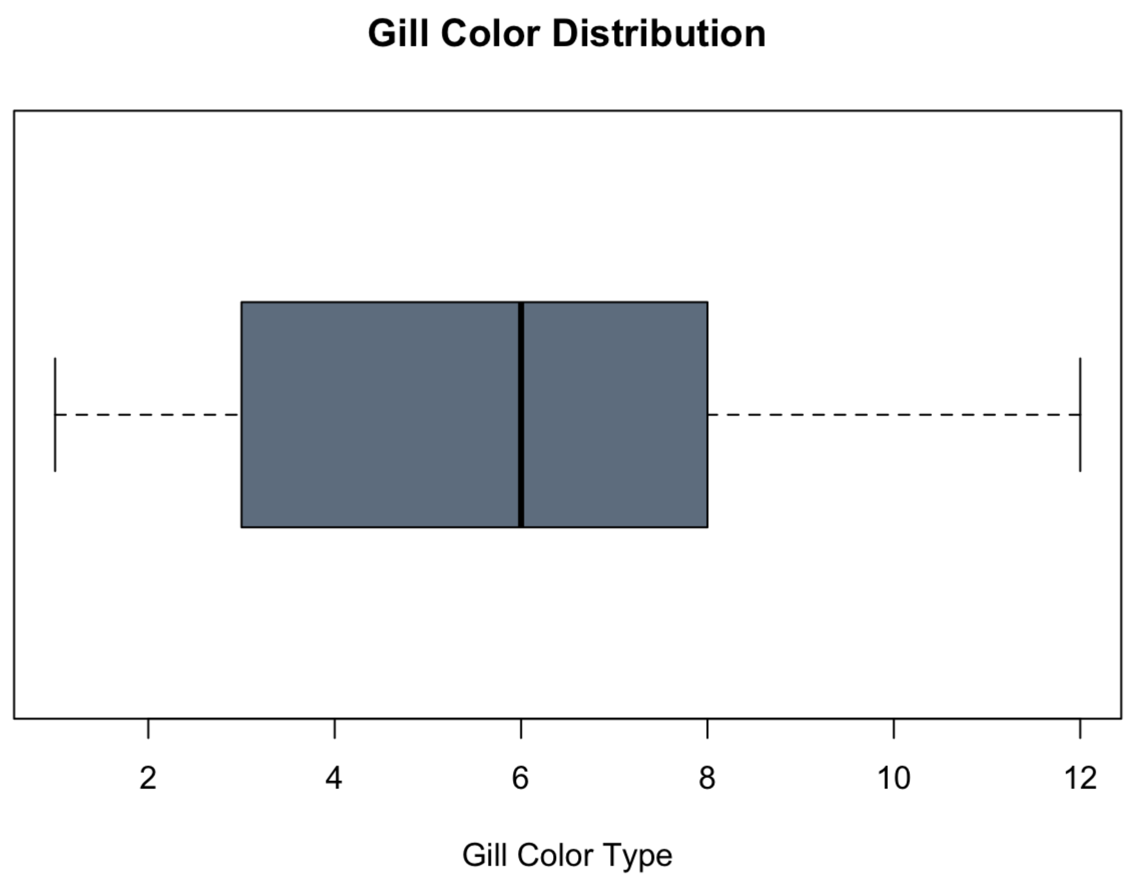
```
boxplot(train$gill.size, col="slategray", horizontal=TRUE, xlab="Gill Size Type", main
="Gill Size Distribution")
```



```
boxplot(train$gill.spacing, col="slategray", horizontal=TRUE, xlab="Gill Spacing Type",
main="Gill Spacing Distribution")
```

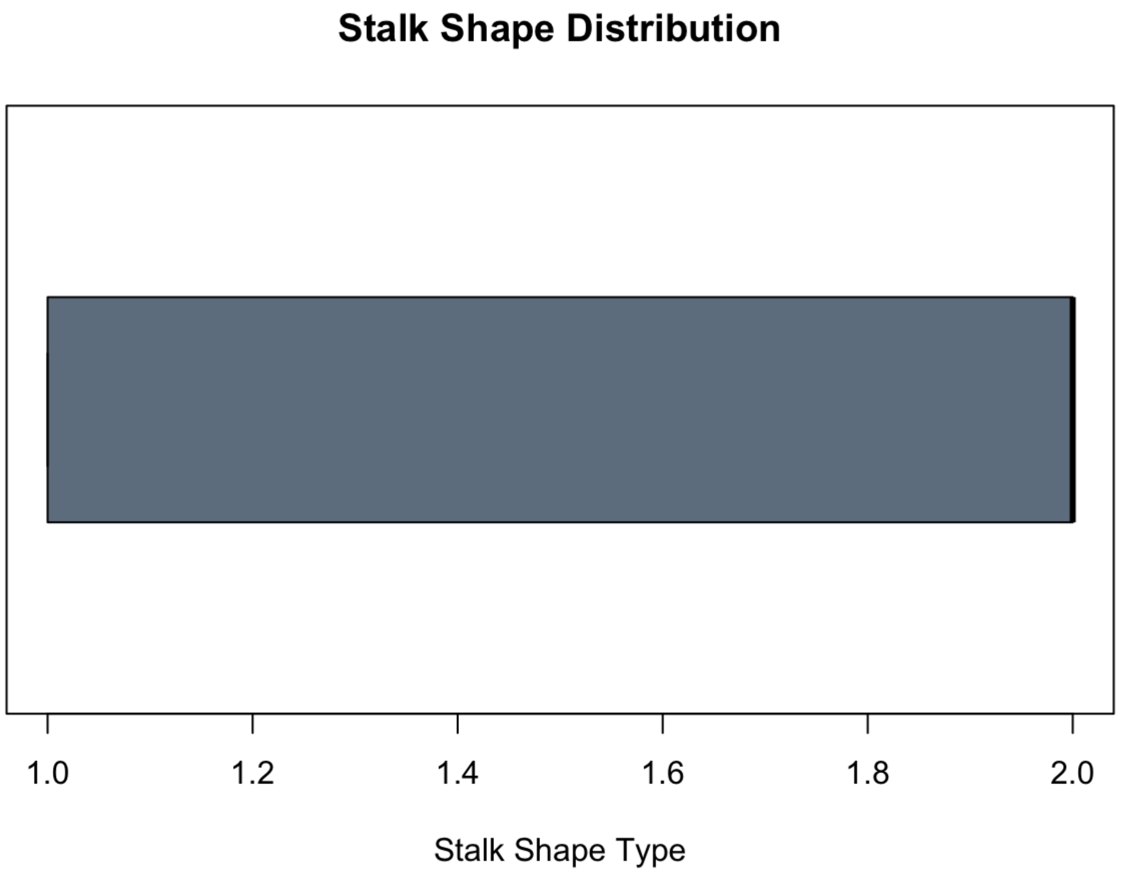


```
boxplot(train$gill.color, col="slategray", horizontal=TRUE, xlab="Gill Color Type", main
="Gill Color Distribution")
```

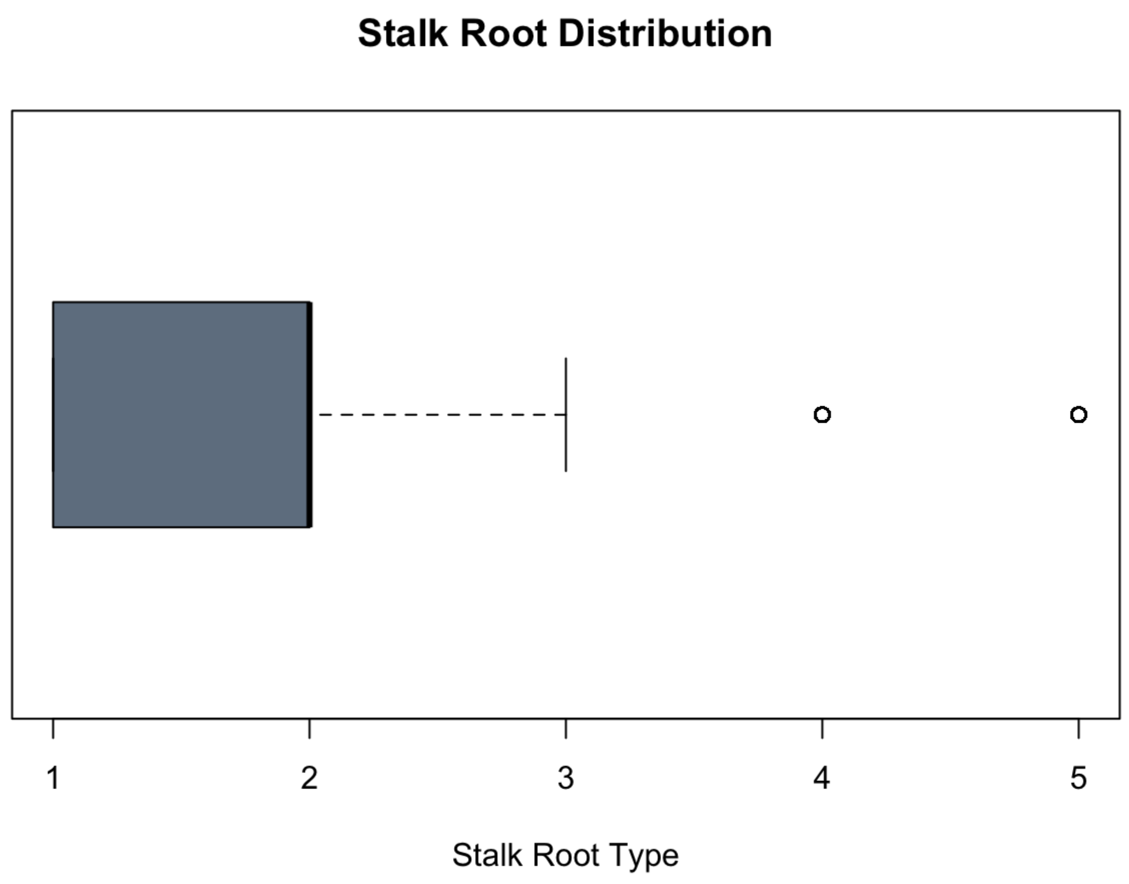


Next, let’s get a sense of the distribution of the stalk shape, root, surface above the ring, surface below the ring, color above the ring, and color below the ring.

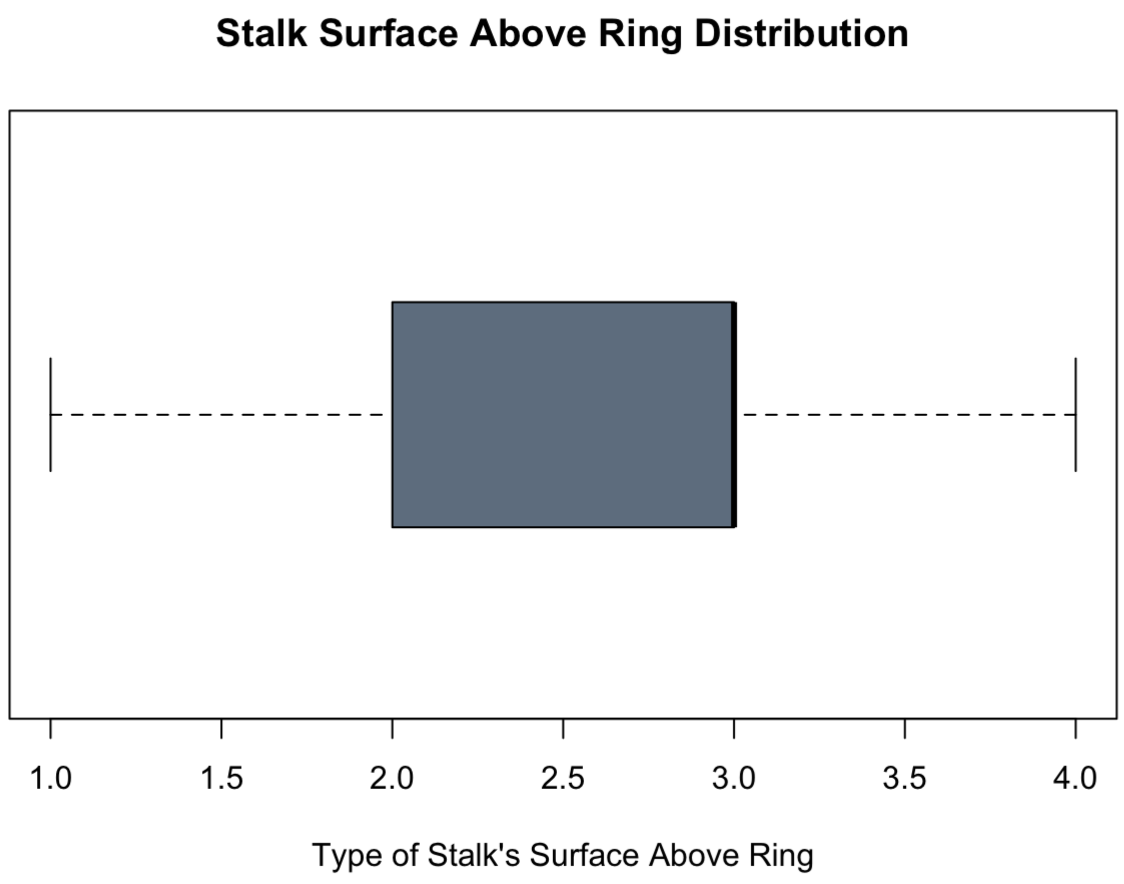
```
boxplot(train$stalk.shape, col="slategray", horizontal=TRUE, xlab="Stalk Shape Type", ma
in="Stalk Shape Distribution")
```



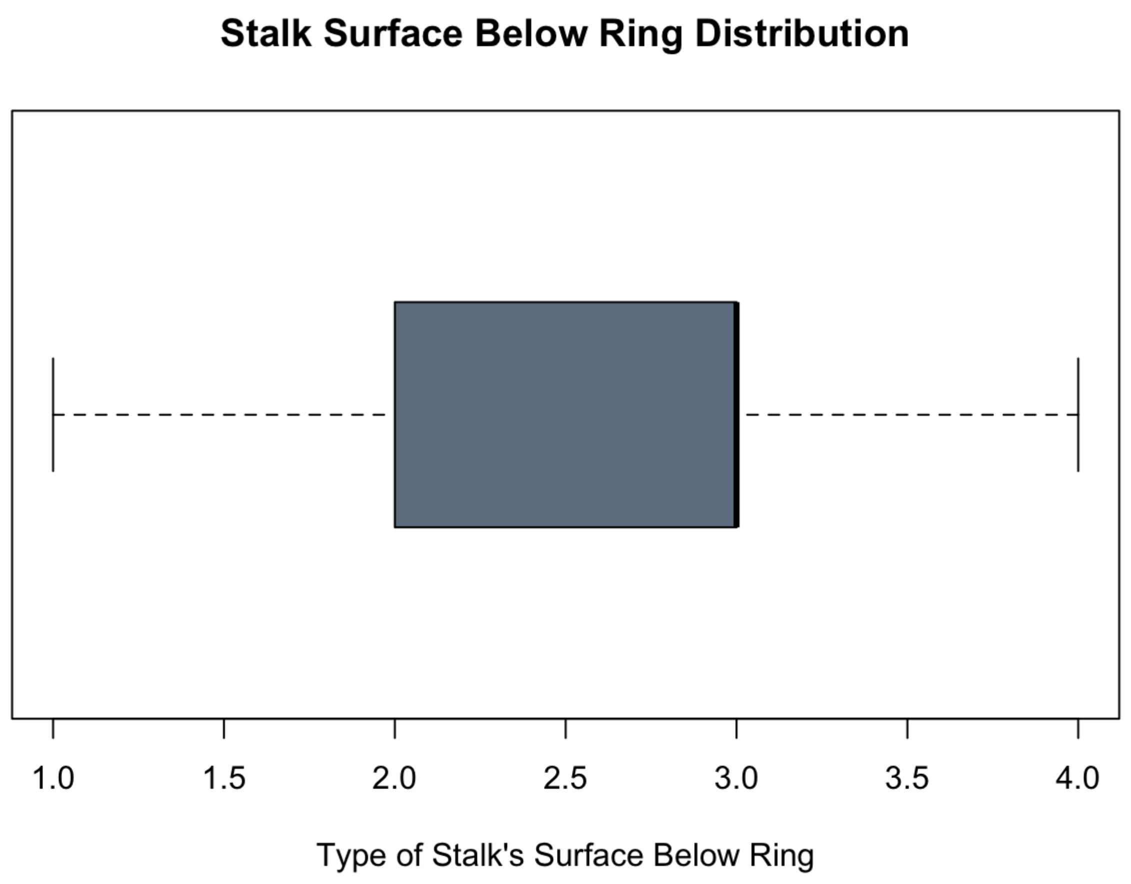
```
boxplot(train$stalk.root, col="slategray", horizontal=TRUE, xlab="Stalk Root Type", main
="Stalk Root Distribution")
```



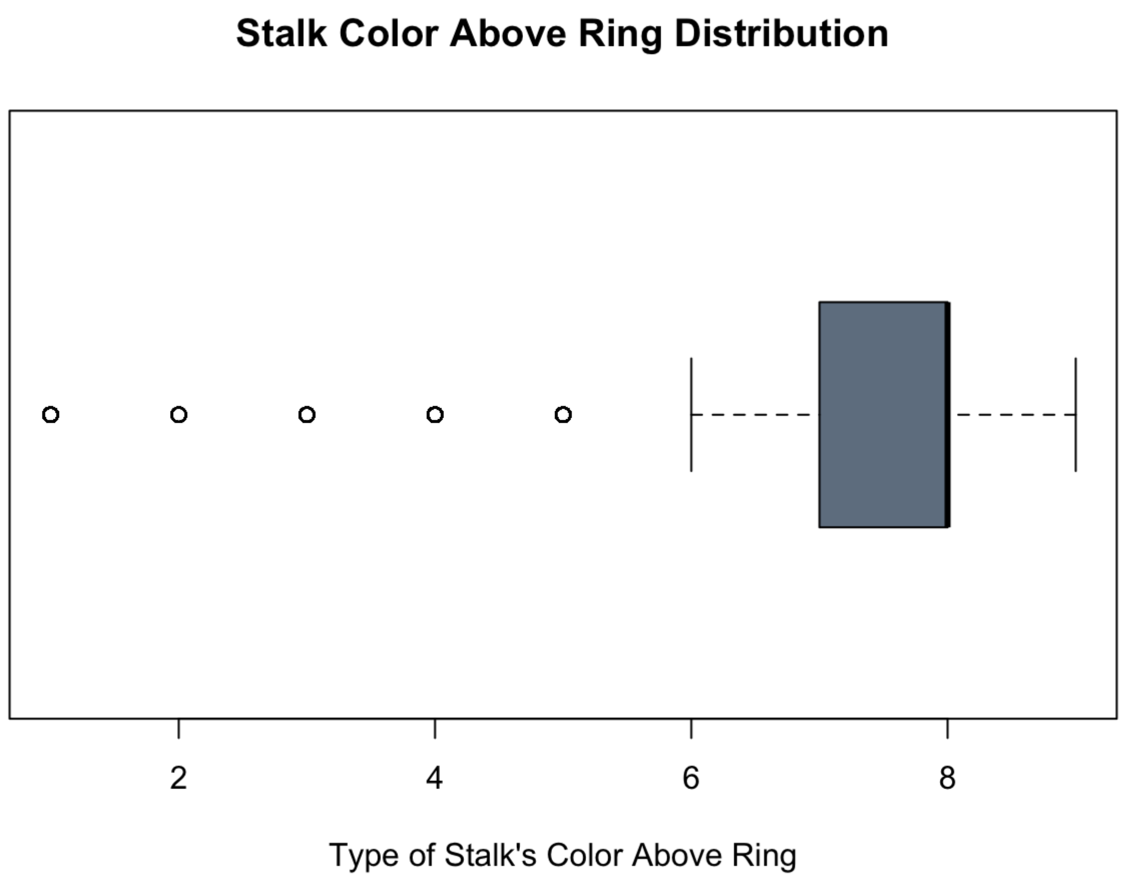
```
boxplot(train$stalk.surface.above.ring, col="slategray", horizontal=TRUE, xlab="Type of
Stalk's Surface Above Ring", main="Stalk Surface Above Ring Distribution")
```



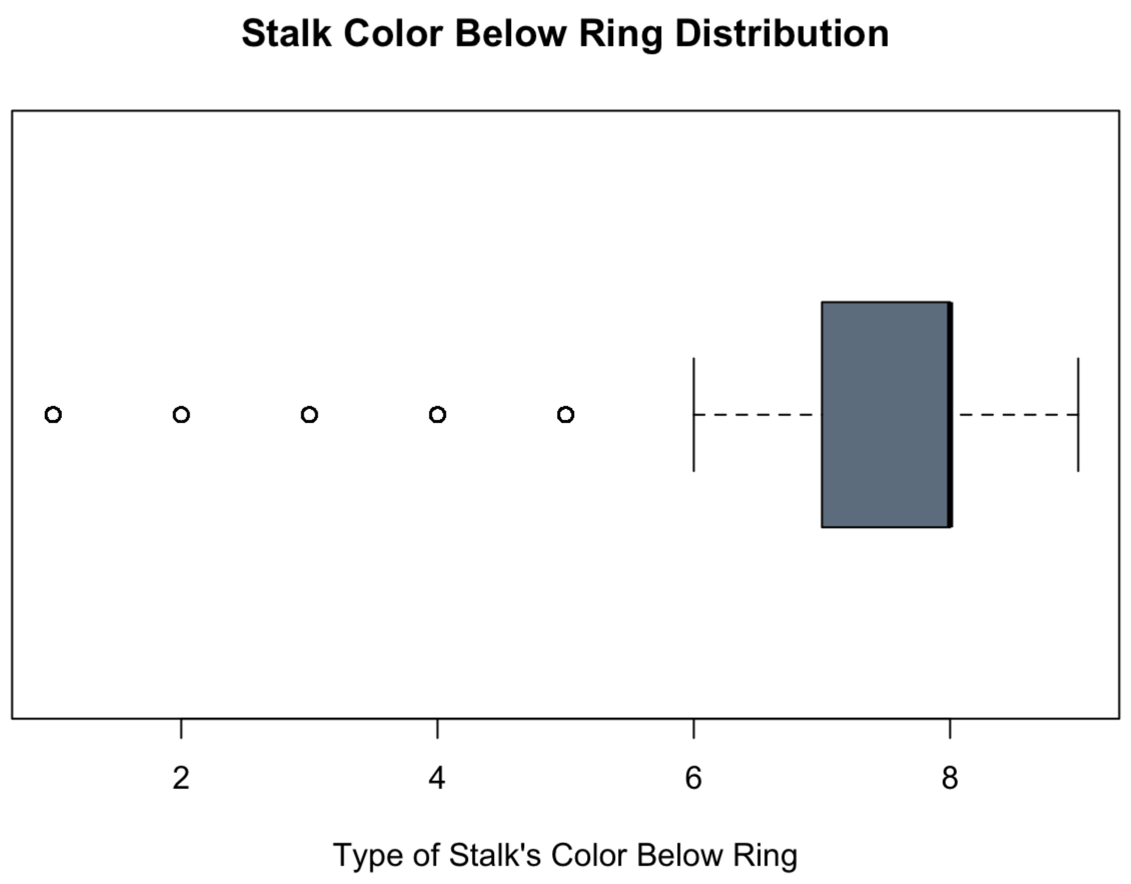
```
boxplot(train$stalk.surface.below.ring, col="slategray", horizontal=TRUE, xlab="Type of
Stalk's Surface Below Ring", main="Stalk Surface Below Ring Distribution")
```



```
boxplot(train$stalk.color.above.ring, col="slategray", horizontal=TRUE, xlab="Type of St
alk's Color Above Ring", main="Stalk Color Above Ring Distribution")
```



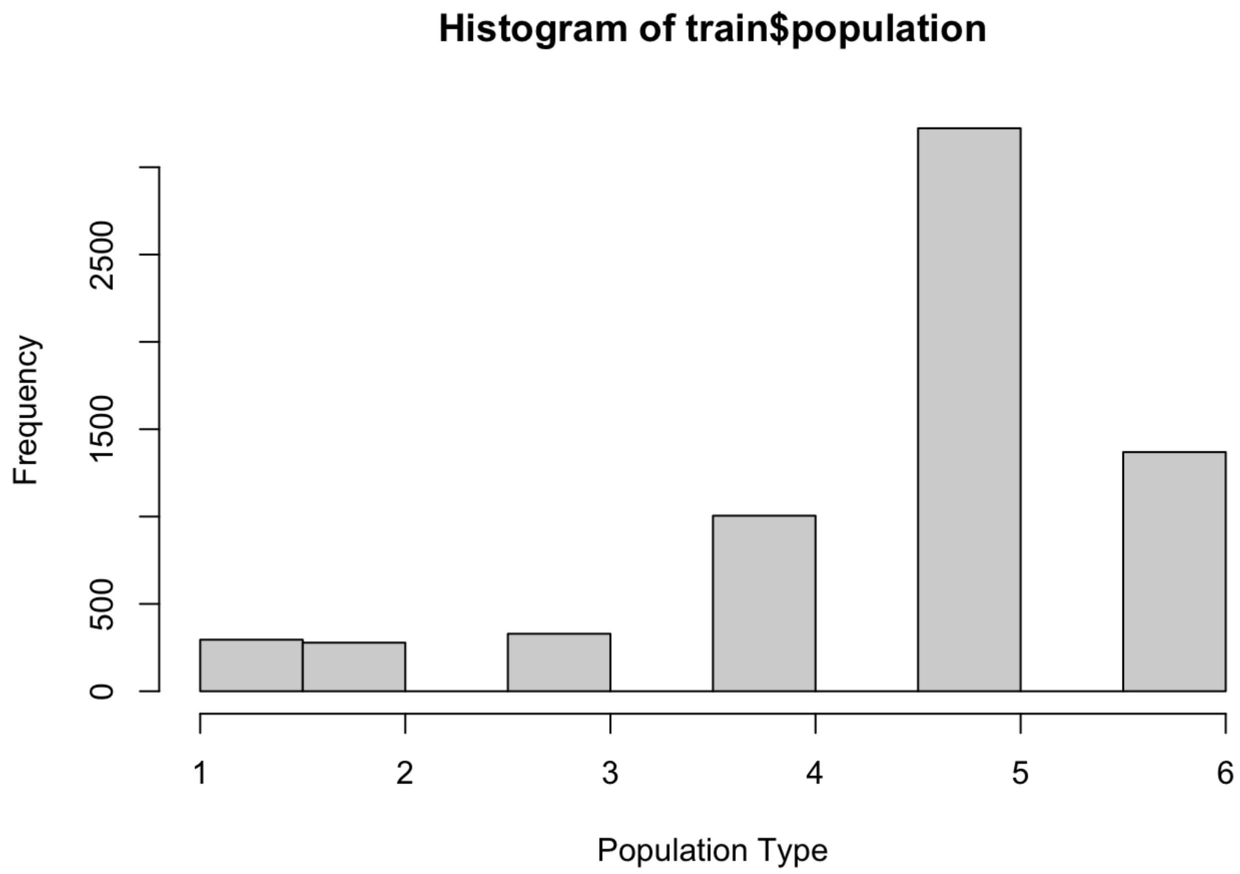
```
boxplot(train$stalk.color.below.ring, col="slategray", horizontal=TRUE, xlab="Type of St
alk's Color Below Ring", main="Stalk Color Below Ring Distribution")
```



Next,

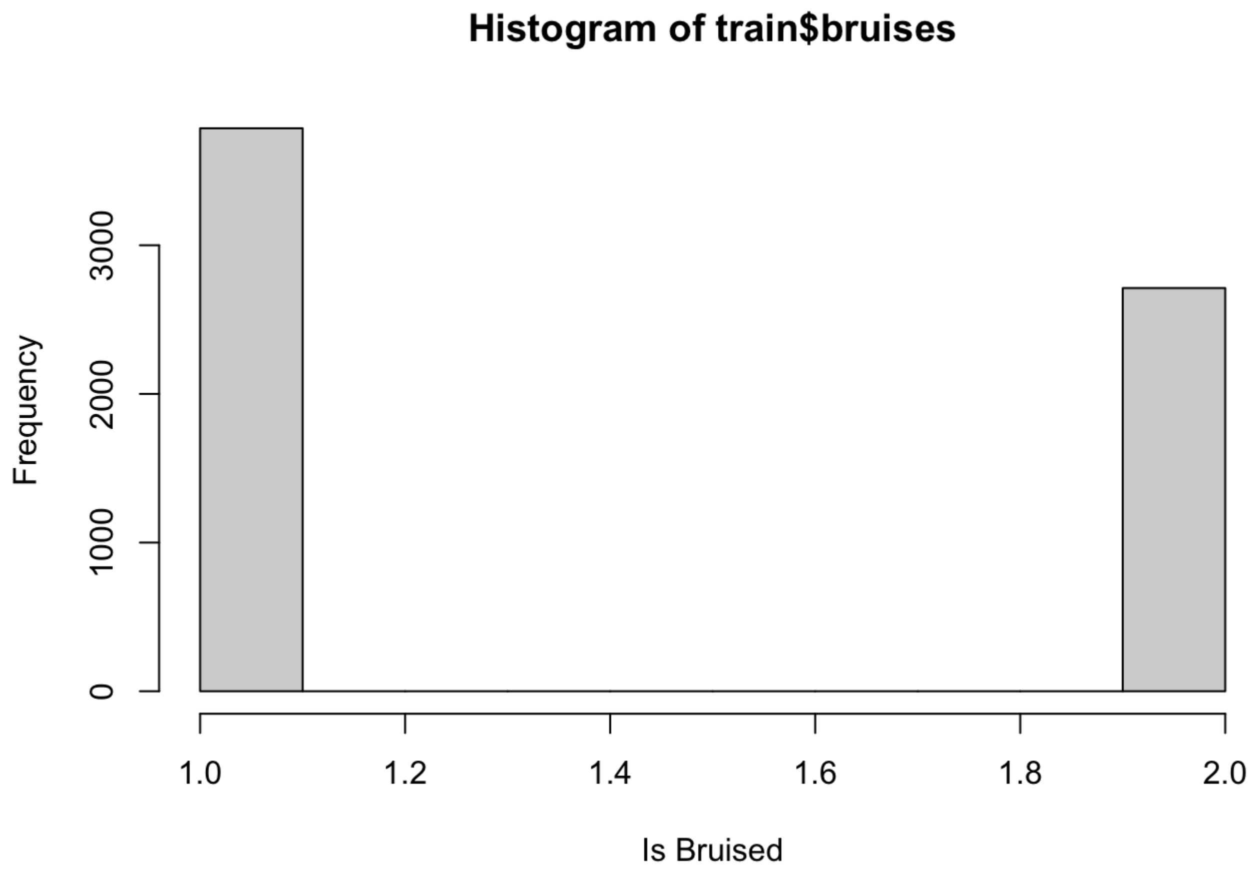
let’s see a histogram of the population type.

```
hist(train$population, xlab="Population Type")
```



Let’s see a histogram of how many mushrooms were bruised in the dataset.

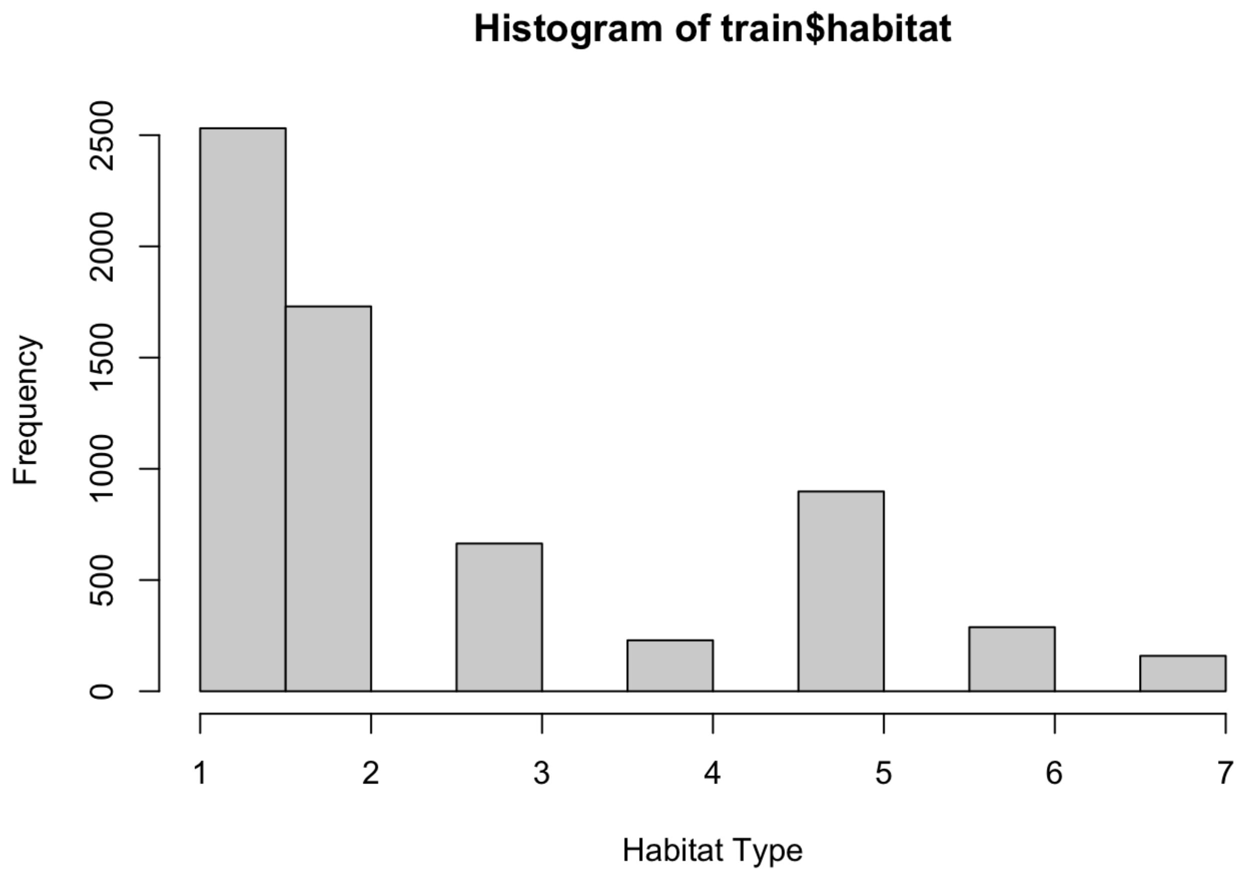
```
hist(train$bruises, xlab="Is Bruised")
```



Now

let's see a histogram corresponding to the habitats.

```
hist(train$habitat, xlab="Habitat Type")
```



Logistic Regression

Next, we will build a logistic regression model and output the summary.

```
glm1 <- logistic_reg(mixture = double(1), penalty = double(1)) %>% set_engine("glmnet")
%>% fit(as.factor(class) ~ cap.shape + cap.surface + cap.color + bruises + odor + gill.a
ttachment + gill.spacing + gill.size + gill.color + stalk.shape + stalk.root + stalk.sur
face.above.ring + stalk.surface.below.ring + stalk.color.above.ring + stalk.color.below.
ring + veil.color + ring.number + ring.type + spore.print.color + population + habitat,
data=train)
tidy(glm1)
```

term <chr>	estimate <dbl>	penalty <dbl>
(Intercept)	3.477554964	0
cap.shape	0.011082585	0
cap.surface	0.193129480	0
cap.color	-0.011892278	0
bruises	-0.974431625	0
odor	-0.175635674	0

term <chr>	estimate <dbl>	penalty <dbl>
gill.attachment	0.495069240	0
gill.spacing	-2.053766378	0
gill.size	2.394967680	0
gill.color	-0.115591055	0

1-10 of 22 rows Previous **1** 2 3 Next

Now, let's run predictions on our data and verify the accuracy of the logistic regression model.

```
pred_class <- predict(glm1,
                      new_data = test,
                      type = "class")
pred_proba <- predict(glm1,
                      new_data = test,
                      type = "prob")
results <- test %>%
           select(class) %>%
           bind_cols(pred_class, pred_proba)

accuracy(results, truth = as.factor(class), estimate = .pred_class)
```

.metric <chr>	.estimator <chr>	.estimate <dbl>
accuracy	binary	0.9261538

1 row

kNN Regression

Next, we will build a kNN regression model. First, we will normalize the data and create a train and test label with the class target variable.

```
normalize <- function(x) { return ((x - min(x)) / (max(x) - min(x))) }
knn_train_df <- data.frame(lapply(new_train_df[1:22], normalize))
knn_train_df <- knn_train_df[-16]
knn_test_df <- data.frame(lapply(new_test_df[1:22], normalize))
knn_test_df <- knn_test_df[-16]
knn_train_labels <- mushrooms[i, 1]
knn_test_labels <- mushrooms[-i, 1]
```

Now, we will train the model and check the accuracy of the predicted values.

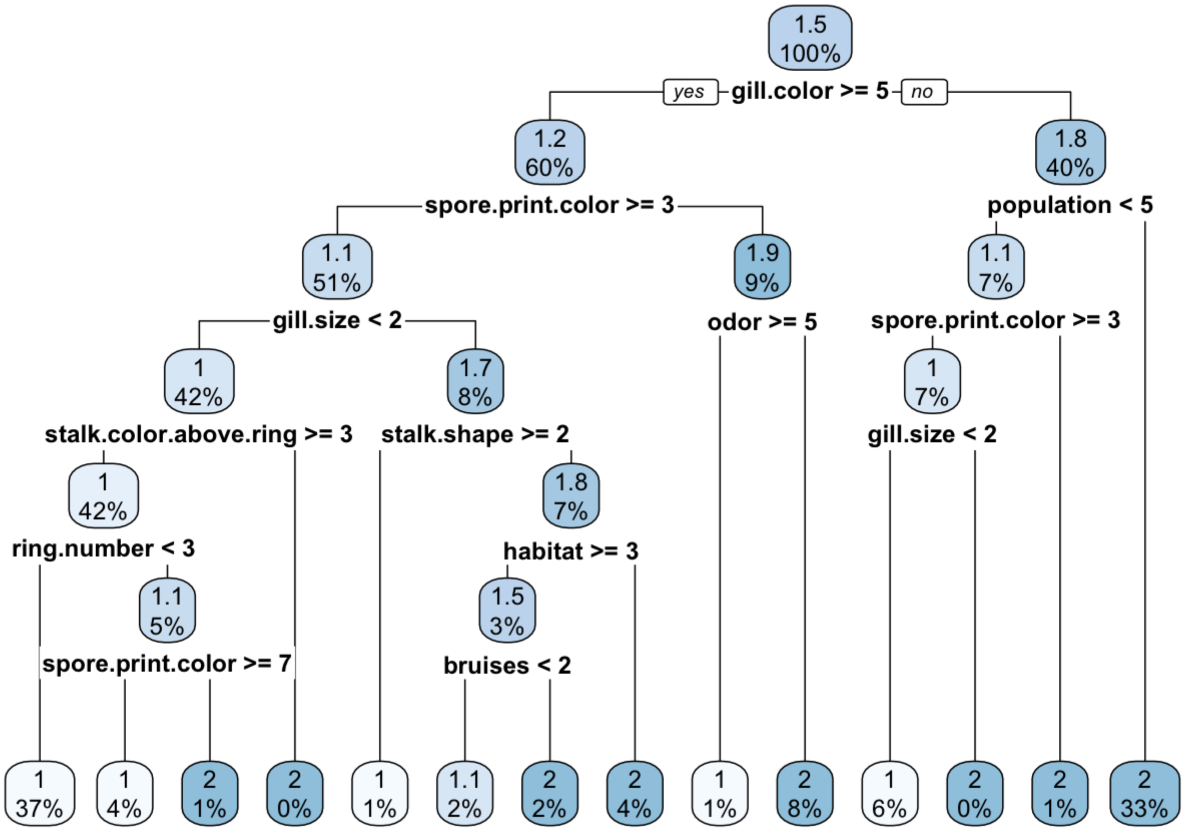
```
knn_predictions <- knn(train=knn_train_df, test=knn_test_df, cl=knn_train_labels, k=40)
CrossTable(x=knn_test_labels, y=knn_predictions, prop.chisq=FALSE)
```

```
## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
## 
## 
## Total Observations in Table:  1625 
## 
## 
##                 | knn_predictions 
## knn_test_labels |         e |         p | Row Total | 
## ----------------|-----------|-----------|-----------|
##               e |       829 |         2 |       831 | 
##                 |     0.998 |     0.002 |     0.511 | 
##                 |     0.999 |     0.003 |           | 
##                 |     0.510 |     0.001 |           | 
## ----------------|-----------|-----------|-----------|
##               p |         1 |       793 |       794 | 
##                 |     0.001 |     0.999 |     0.489 | 
##                 |     0.001 |     0.997 |           | 
##                 |     0.001 |     0.488 |           | 
## ----------------|-----------|-----------|-----------|
##    Column Total |       830 |       795 |      1625 | 
##                 |     0.511 |     0.489 |           | 
## ----------------|-----------|-----------|-----------|
## 
## 
```

Decision Tree Regression

Next, we will build a decision tree regression model, plot the decision tree, and output the summary.

```
dtree <- rpart(formula=as.factor(class) ~ cap.shape + cap.surface + cap.color + bruises
+ odor + gill.attachment + gill.spacing + gill.size + gill.color + stalk.shape + stalk.r
oot + stalk.surface.above.ring + stalk.surface.below.ring + stalk.color.above.ring + sta
lk.color.below.ring + veil.color + ring.number + ring.type + spore.print.color + populat
ion + habitat, data=train, method="anova")
rpart.plot(dtree)
```



```
summary(dtree)
```

```
## Call:
## rpart(formula = as.factor(class) ~ cap.shape + cap.surface +
##     cap.color + bruises + odor + gill.attachment + gill.spacing +
##     gill.size + gill.color + stalk.shape + stalk.root + stalk.surface.above.ring +
##     stalk.surface.below.ring + stalk.color.above.ring + stalk.color.below.ring +
##     veil.color + ring.number + ring.type + spore.print.color +
##     population + habitat, data = train, method = "anova")
##   n= 6499
##
##            CP nsplit  rel error     xerror         xstd
## 1  0.33847636      0 1.00000000 1.00025592 0.0009786708
## 2  0.17911052      1 0.66152364 0.66185140 0.0116241597
## 3  0.16584867      2 0.48241312 0.48284032 0.0103662974
## 4  0.11958461      3 0.31656446 0.31688187 0.0107921921
## 5  0.04070978      4 0.19697984 0.19746589 0.0083315869
## 6  0.02652549      5 0.15627007 0.15670521 0.0074217978
## 7  0.02152007      6 0.12974458 0.13008771 0.0074412833
## 8  0.02064290      8 0.08670444 0.10311990 0.0070230605
## 9  0.01726147      9 0.06606154 0.06623857 0.0061126293
## 10 0.01119939     10 0.04880007 0.04896949 0.0052860623
## 11 0.01004771     11 0.03760067 0.04636026 0.0051715733
## 12 0.01000000     13 0.01750525 0.03250427 0.0043452488
##
## Variable importance
##        spore.print.color               gill.color                     odor
##                       15                       13                       11
##               stalk.root                ring.type                  bruises
##                       11                       10                        8
##               population                gill.size             gill.spacing
##                        7                        6                        4
##              ring.number   stalk.color.below.ring   stalk.color.above.ring
##                        3                        2                        2
##                  habitat stalk.surface.below.ring              stalk.shape
##                        2                        1                        1
##               veil.color                cap.color          gill.attachment
##                        1                        1                        1
##              cap.surface stalk.surface.above.ring
##                        1                        1
##
## Node number 1: 6499 observations,    complexity param=0.3384764
##   mean=1.480382, MSE=0.2496151
##   left son=2 (3872 obs) right son=3 (2627 obs)
##   Primary splits:
##       gill.color < 4.5 to the right, improve=0.3384764, (0 missing)
##       ring.type  < 4.5 to the right, improve=0.2940224, (0 missing)
##       odor       < 3.5 to the right, improve=0.2883337, (0 missing)
##       gill.size  < 1.5 to the left,  improve=0.2869113, (0 missing)
##       bruises    < 1.5 to the right, improve=0.2513931, (0 missing)
##   Surrogate splits:
##       ring.type         < 3.5 to the right, agree=0.799, adj=0.503, (0 split)
##       spore.print.color < 7.5 to the left,  agree=0.764, adj=0.416, (0 split)
##       stalk.root        < 1.5 to the right, agree=0.760, adj=0.406, (0 split)
##       bruises           < 1.5 to the right, agree=0.737, adj=0.349, (0 split)
##       odor              < 7.5 to the left,  agree=0.734, adj=0.342, (0 split)
##
## Node number 2: 3872 observations,    complexity param=0.1658487
##   mean=1.240961, MSE=0.1828987
##   left son=4 (3292 obs) right son=5 (580 obs)
##   Primary splits:
##       spore.print.color        < 2.5 to the right, improve=0.3799124, (0 missing)
##       odor                     < 3.5 to the right, improve=0.3375228, (0 missing)
##       stalk.surface.above.ring < 2.5 to the right, improve=0.1816445, (0 missing)
##       stalk.shape              < 1.5 to the right, improve=0.1674649, (0 missing)
##       gill.size                < 1.5 to the left,  improve=0.1499810, (0 missing)
##   Surrogate splits:
##       stalk.surface.below.ring < 2.5 to the right, agree=0.885, adj=0.229, (0 split)
##       stalk.color.below.ring   < 1.5 to the right, agree=0.883, adj=0.217, (0 split)
##       stalk.color.above.ring   < 1.5 to the right, agree=0.880, adj=0.200, (0 split)
##       odor                     < 3.5 to the right, agree=0.876, adj=0.174, (0 split)
##       stalk.surface.above.ring < 2.5 to the right, agree=0.862, adj=0.079, (0 split)
##
## Node number 3: 2627 observations,    complexity param=0.1791105
##   mean=1.83327, MSE=0.1389312
##   left son=6 (474 obs) right son=7 (2153 obs)
##   Primary splits:
##       population   < 4.5 to the left,  improve=0.7961203, (0 missing)
##       stalk.root   < 2.5 to the right, improve=0.5630634, (0 missing)
##       gill.spacing < 1.5 to the right, improve=0.4478229, (0 missing)
##       ring.number  < 2.5 to the right, improve=0.2910835, (0 missing)
##       gill.color   < 1.5 to the right, improve=0.2131476, (0 missing)
##   Surrogate splits:
##       gill.spacing < 1.5 to the right, agree=0.910, adj=0.502, (0 split)
##       stalk.root   < 2.5 to the right, agree=0.909, adj=0.494, (0 split)
##       bruises      < 1.5 to the right, agree=0.882, adj=0.344, (0 split)
##       ring.type    < 4   to the right, agree=0.879, adj=0.327, (0 split)
##       ring.number  < 2.5 to the right, agree=0.876, adj=0.310, (0 split)
##
## Node number 4: 3292 observations,    complexity param=0.1195846
##   mean=1.130316, MSE=0.1133337
##   left son=8 (2750 obs) right son=9 (542 obs)
##   Primary splits:
##       gill.size              < 1.5 to the left,  improve=0.51996480, (0 missing)
##       odor                   < 6.5 to the left,  improve=0.44779500, (0 missing)
##       stalk.shape            < 1.5 to the right, improve=0.20478160, (0 missing)
##       stalk.color.above.ring < 2.5 to the right, improve=0.05931233, (0 missing)
##       stalk.color.below.ring < 2.5 to the right, improve=0.05931233, (0 missing)
##   Surrogate splits:
##       odor                   < 6.5 to the left,  agree=0.898, adj=0.382, (0 split)
##       habitat                < 5.5 to the left,  agree=0.858, adj=0.138, (0 split)
##       stalk.color.below.ring < 8.5 to the left,  agree=0.841, adj=0.035, (0 split)
##       stalk.color.above.ring < 8.5 to the left,  agree=0.837, adj=0.011, (0 split)
##       veil.color             < 3.5 to the left,  agree=0.837, adj=0.011, (0 split)
##
## Node number 5: 580 observations,    complexity param=0.04070978
##   mean=1.868966, MSE=0.1138644 
##   left son=10 (76 obs) right son=11 (504 obs)
##   Primary splits:
##       odor              < 4.5 to the right, improve=1.0000000, (0 missing)
##       stalk.root        < 1.5 to the left,  improve=1.0000000, (0 missing)
##       gill.attachment   < 1.5 to the left,  improve=0.5177069, (0 missing)
##       veil.color        < 2.5 to the left,  improve=0.5177069, (0 missing)
##       spore.print.color < 1.5 to the left,  improve=0.5177069, (0 missing)
##   Surrogate splits:
##       stalk.root        < 1.5 to the left,  agree=1.000, adj=1.000, (0 split)
##       gill.attachment   < 1.5 to the left,  agree=0.941, adj=0.553, (0 split)
##       veil.color        < 2.5 to the left,  agree=0.941, adj=0.553, (0 split)
##       spore.print.color < 1.5 to the left,  agree=0.941, adj=0.553, (0 split)
##       gill.size         < 1.5 to the right, agree=0.928, adj=0.447, (0 split)
## 
## Node number 6: 474 observations,    complexity param=0.0206429
##   mean=1.124473, MSE=0.1089792 
##   left son=12 (434 obs) right son=13 (40 obs)
##   Primary splits:
##       spore.print.color < 2.5 to the right, improve=0.6482856, (0 missing)
##       odor              < 3.5 to the right, improve=0.4760694, (0 missing)
##       gill.size         < 1.5 to the left,  improve=0.2937232, (0 missing)
##       habitat           < 1.5 to the right, improve=0.2937232, (0 missing)
##       population        < 3.5 to the left,  improve=0.1613807, (0 missing)
## 
## Node number 7: 2153 observations
##   mean=1.989317, MSE=0.01056865 
## 
## Node number 8: 2750 observations,    complexity param=0.01726147
##   mean=1.022545, MSE=0.02203716 
##   left son=16 (2721 obs) right son=17 (29 obs)
##   Primary splits:
##       stalk.color.above.ring < 2.5 to the right, improve=0.4620692, (0 missing)
##       stalk.color.below.ring < 2.5 to the right, improve=0.4620692, (0 missing)
##       ring.number            < 1.5 to the right, improve=0.4620692, (0 missing)
##       spore.print.color      < 5.5 to the left,  improve=0.1434176, (0 missing)
##       cap.color              < 2.5 to the right, improve=0.1281378, (0 missing)
##   Surrogate splits:
##       stalk.color.below.ring < 2.5 to the right, agree=1, adj=1, (0 split)
##       ring.number            < 1.5 to the right, agree=1, adj=1, (0 split)
## 
## Node number 9: 542 observations,    complexity param=0.02652549
##   mean=1.677122, MSE=0.2186279 
##   left son=18 (80 obs) right son=19 (462 obs)
##   Primary splits:
##       stalk.shape  < 1.5 to the right, improve=0.3631416, (0 missing)
##       odor         < 6.5 to the left,  improve=0.2946439, (0 missing)
##       gill.spacing < 1.5 to the right, improve=0.2259592, (0 missing)
##       population   < 4.5 to the right, improve=0.2216658, (0 missing)
##       cap.surface  < 1.5 to the left,  improve=0.1891988, (0 missing)
##   Surrogate splits:
##       odor      < 1.5 to the left,  agree=0.934, adj=0.55, (0 split)
##       cap.color < 9.5 to the right, agree=0.889, adj=0.25, (0 split)
## 
## Node number 10: 76 observations
##   mean=1, MSE=0 
## 
## Node number 11: 504 observations
##   mean=2, MSE=0 
## 
## Node number 12: 434 observations,    complexity param=0.01119939
##   mean=1.043779, MSE=0.04186222 
##   left son=24 (415 obs) right son=25 (19 obs)
##   Primary splits:
##       gill.size  < 1.5 to the left,  improve=1.00000000, (0 missing)
##       habitat    < 1.5 to the right, improve=1.00000000, (0 missing)
##       odor       < 3   to the right, improve=0.24218610, (0 missing)
##       population < 3.5 to the left,  improve=0.06339203, (0 missing)
##       stalk.root < 2.5 to the right, improve=0.06220534, (0 missing)
##   Surrogate splits:
##       habitat < 1.5 to the right, agree=1, adj=1, (0 split)
## 
## Node number 13: 40 observations
##   mean=2, MSE=0 
## 
## Node number 16: 2721 observations,    complexity param=0.01004771
##   mean=1.012128, MSE=0.01198081 
##   left son=32 (2407 obs) right son=33 (314 obs)
##   Primary splits:
##       ring.number       < 2.5 to the left,  improve=0.09410899, (0 missing)
##       cap.color         < 1.5 to the right, improve=0.08302279, (0 missing)
##       spore.print.color < 5.5 to the left,  improve=0.08262416, (0 missing)
##       stalk.shape       < 1.5 to the right, improve=0.02484003, (0 missing)
##       cap.shape         < 2   to the right, improve=0.02187676, (0 missing)
##   Surrogate splits:
##       spore.print.color      < 5.5 to the left,  agree=0.986, adj=0.879, (0 split)
##       stalk.root             < 1.5 to the right, agree=0.929, adj=0.382, (0 split)
##       habitat                < 6   to the left,  agree=0.914, adj=0.252, (0 split)
##       stalk.color.below.ring < 3.5 to the right, agree=0.899, adj=0.124, (0 split)
##       cap.color              < 2.5 to the right, agree=0.898, adj=0.115, (0 split)
## 
## Node number 17: 29 observations
##   mean=2, MSE=0 
## 
## Node number 18: 80 observations
##   mean=1, MSE=0 
## 
## Node number 19: 462 observations,    complexity param=0.02152007
##   mean=1.794372, MSE=0.163345 
##   left son=38 (206 obs) right son=39 (256 obs)
##   Primary splits:
##       habitat                < 2.5 to the right, improve=0.3216846, (0 missing)
##       population             < 5.5 to the right, improve=0.2492360, (0 missing)
##       cap.surface            < 1.5 to the left,  improve=0.2451060, (0 missing)
##       stalk.color.below.ring < 6.5 to the left,  improve=0.2391844, (0 missing)
##       cap.color              < 5.5 to the left,  improve=0.2214307, (0 missing)
##   Surrogate splits:
##       odor       < 4   to the right, agree=0.710, adj=0.350, (0 split)
##       stalk.root < 2.5 to the right, agree=0.667, adj=0.252, (0 split)
##       cap.shape  < 5.5 to the left,  agree=0.660, adj=0.238, (0 split)
##       cap.color  < 5.5 to the left,  agree=0.652, adj=0.218, (0 split)
##       population < 5.5 to the right, agree=0.615, adj=0.136, (0 split)
##
## Node number 24: 415 observations
##   mean=1, MSE=0
##
## Node number 25: 19 observations
##   mean=2, MSE=0
##
## Node number 32: 2407 observations
##   mean=1, MSE=0
##
## Node number 33: 314 observations,    complexity param=0.01004771
##   mean=1.105096, MSE=0.09405047
##   left son=66 (281 obs) right son=67 (33 obs)
##   Primary splits:
##       spore.print.color < 7   to the right, improve=1.0000000, (0 missing)
##       stalk.root        < 1.5 to the left,  improve=0.3742349, (0 missing)
##       population        < 4.5 to the left,  improve=0.3742349, (0 missing)
##       cap.shape         < 3.5 to the right, improve=0.1577522, (0 missing)
##       bruises           < 1.5 to the left,  improve=0.1440903, (0 missing)
##
## Node number 38: 206 observations,    complexity param=0.02152007
##   mean=1.538835, MSE=0.2484918
##   left son=76 (101 obs) right son=77 (105 obs)
##   Primary splits:
##       bruises     < 1.5 to the left,  improve=0.8897511, (0 missing)
##       odor        < 6.5 to the left,  improve=0.7918666, (0 missing)
##       cap.surface < 1.5 to the left,  improve=0.6413138, (0 missing)
##       cap.color   < 7   to the left,  improve=0.3435068, (0 missing)
##       population  < 4.5 to the right, improve=0.3195195, (0 missing)
##   Surrogate splits:
##       odor         < 6.5 to the left,  agree=0.971, adj=0.941, (0 split)
##       cap.surface  < 1.5 to the left,  agree=0.864, adj=0.723, (0 split)
##       cap.color    < 4.5 to the left,  agree=0.733, adj=0.455, (0 split)
##       ring.type    < 3   to the left,  agree=0.733, adj=0.455, (0 split)
##       gill.spacing < 1.5 to the right, agree=0.704, adj=0.396, (0 split)
##
## Node number 39: 256 observations
##   mean=2, MSE=0
##
## Node number 66: 281 observations
##   mean=1, MSE=0
##
## Node number 67: 33 observations
##   mean=2, MSE=0
```

```
##
## Node number 76: 101 observations
##   mean=1.059406, MSE=0.05587687
##
## Node number 77: 105 observations
##   mean=2, MSE=0
```

Analysis

For the logistic regression model, we can see that the estimate column represents the coefficients for the predictor variables. The penalty column in the summary then represents if any regularization was applied to the model, which there was not. So the model could accurately predict around 93% of the observations in the test set. For the kNN model, we are given a confusion matrix output where the diagonal areas represent the True Positive and True Negative predictions for the predicted labels. In contrast, the other areas represent False Positive and False Negative. The kNN model correctly predicted the e class 829 times (true positive) but had two false negatives. Using this data, we can calculate the accuracy with the equation (TP + TN) / (TP + TN + FP + FN), meaning the model had a 99.8% accuracy. Finally, for the decision tree regression model, the summary tells us that the spore print color, gill color, stalk root, ring type, and odor were the best variables to predict the mushroom class type correctly. Let us look at the summary of node 1. We notice that many observations were made after splitting the left and right subtree but also had a high mean and mean squared error. This indicates that the model's accuracy might need to consider other factors.

Sources

https://www.kaggle.com/datasets/uciml/mushroom-classification (https://www.kaggle.com/datasets/uciml/mushroom-classification) https://www.datacamp.com/tutorial/logistic-regression-R (https://www.datacamp.com/tutorial/logistic-regression-R) https://www.kaggle.com/code/nicktp/mushroom-classification-with-r (https://www.kaggle.com/code/nicktp/mushroom-classification-with-r) https://www.analyticsvidhya.com/blog/2015/08/learning-concept-knn-algorithms-programming/ (https://www.analyticsvidhya.com/blog/2015/08/learning-concept-knn-algorithms-programming/) https://www.edureka.co/blog/knn-algorithm-in-r/ (https://www.edureka.co/blog/knn-algorithm-in-r/) https://www.datacamp.com/tutorial/decision-trees-R (https://www.datacamp.com/tutorial/decision-trees-R) https://uc-r.github.io/regression_trees (https://uc-r.github.io/regression_trees)