Project 4: Classification Analysis

DA 410

Marjorie Blanco

Problem 9.7 (c)

Use R built-in function (e.g. knn) to do 9.7 (c), try different k values and find out the best solution.

Do some EDA at the very beginning and make a clean explanation of how and why you choose your final model.

Four measurements were made on two species of flea beetles (Lubischew 1962). The variables are:

y1 = distance of transverse groove from posterior border of prothorax (μm),

y2 = length of elytra (in .01 mm),

y3 = length of second antennal joint (μm),

y4 = length of third antennal joint (μm).

Table 1: Summary Statistics of the Flea Beetles data set

```
##        y1              y2              y3              y4
##  Min.   :158.0   Min.   :237.0   Min.   :121.0   Min.   :158.0
##  1st Qu.:177.0   1st Qu.:262.5   1st Qu.:137.5   1st Qu.:187.0
##  Median :184.0   Median :278.0   Median :146.0   Median :197.0
##  Mean   :186.8   Mean   :279.2   Mean   :147.5   Mean   :197.9
##  3rd Qu.:193.5   3rd Qu.:299.0   3rd Qu.:161.0   3rd Qu.:213.0
##  Max.   :221.0   Max.   :317.0   Max.   :184.0   Max.   :235.0
```

From above summary statistics, it shows us that all the attributes have a different range. The data needs to be standardized.

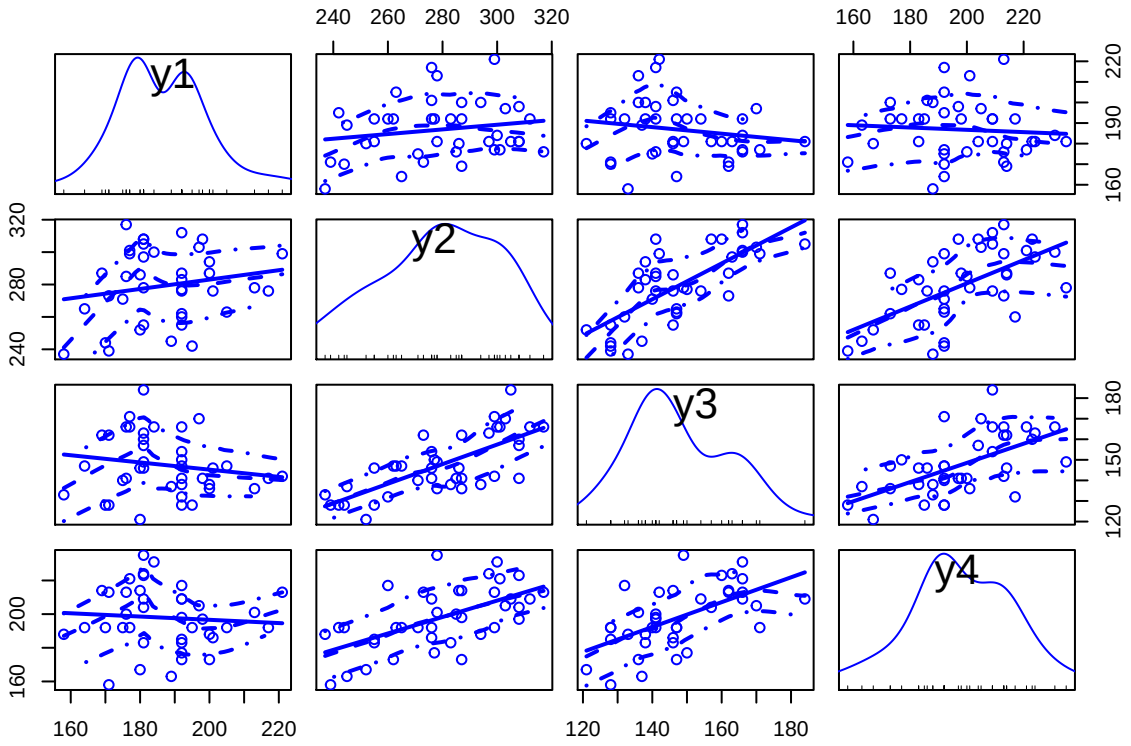
Table 2: Summary Statistics of the Haltica oleracea species

```
##        y1              y2              y3              y4
##  Min.   :170.0   Min.   :239.0   Min.   :121.0   Min.   :158.0
##  1st Qu.:190.5   1st Qu.:253.5   1st Qu.:130.0   1st Qu.:175.0
##  Median :192.0   Median :263.0   Median :138.0   Median :186.0
##  Mean   :194.5   Mean   :267.1   Mean   :137.4   Mean   :185.9
##  3rd Qu.:200.5   3rd Qu.:280.5   3rd Qu.:144.0   3rd Qu.:192.0
##  Max.   :221.0   Max.   :299.0   Max.   :150.0   Max.   :217.0
```

Table 3: Summary Statistics of the Haltica carduorum species

```
##        y1              y2              y3              y4
##  Min.   :158.0   Min.   :237.0   Min.   :133.0   Min.   :188.0
##  1st Qu.:175.8   1st Qu.:277.5   1st Qu.:146.8   1st Qu.:199.2
##  Median :180.5   Median :298.0   Median :161.0   Median :209.0
##  Mean   :179.6   Mean   :290.8   Mean   :157.2   Mean   :209.2
##  3rd Qu.:181.8   3rd Qu.:305.8   3rd Qu.:166.0   3rd Qu.:215.8
##  Max.   :198.0   Max.   :317.0   Max.   :184.0   Max.   :235.0
```

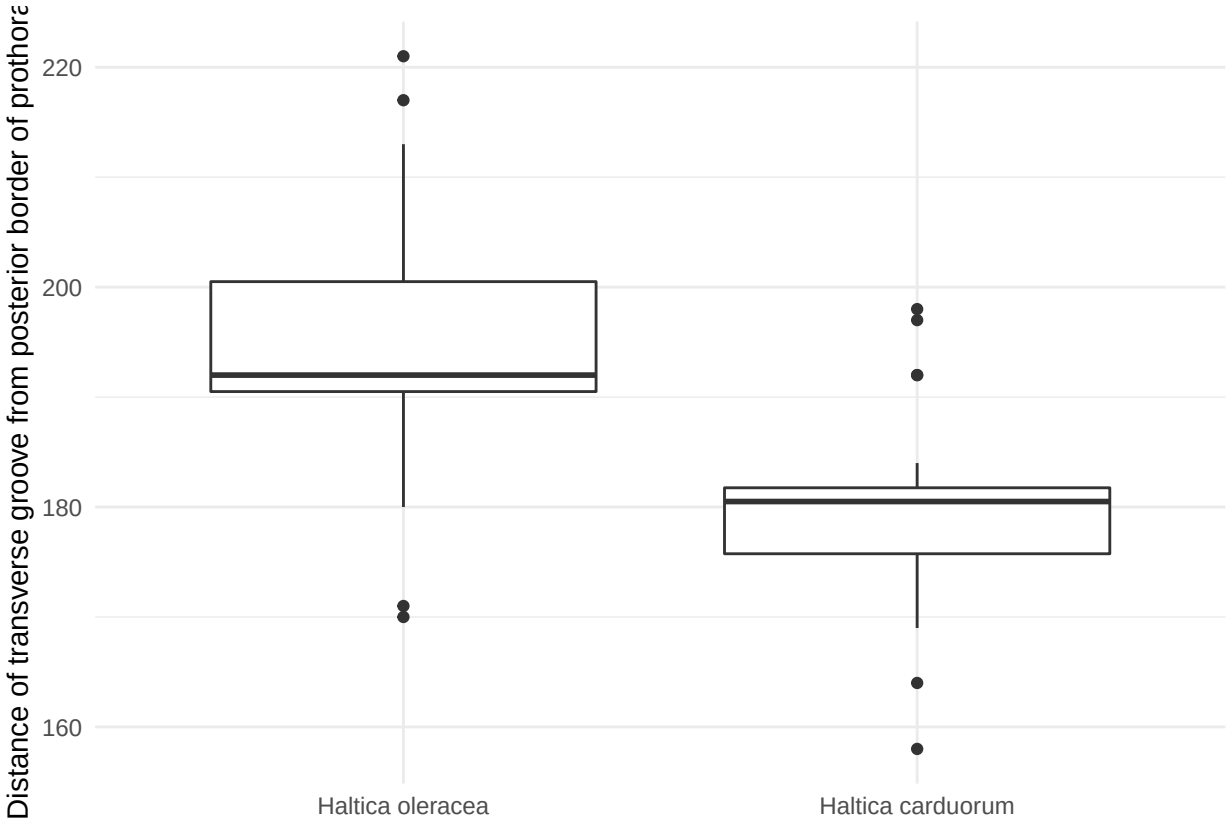
As noted in the introduction section, the measurement unit for length of elytra (mm) is different the the other 3 variables.



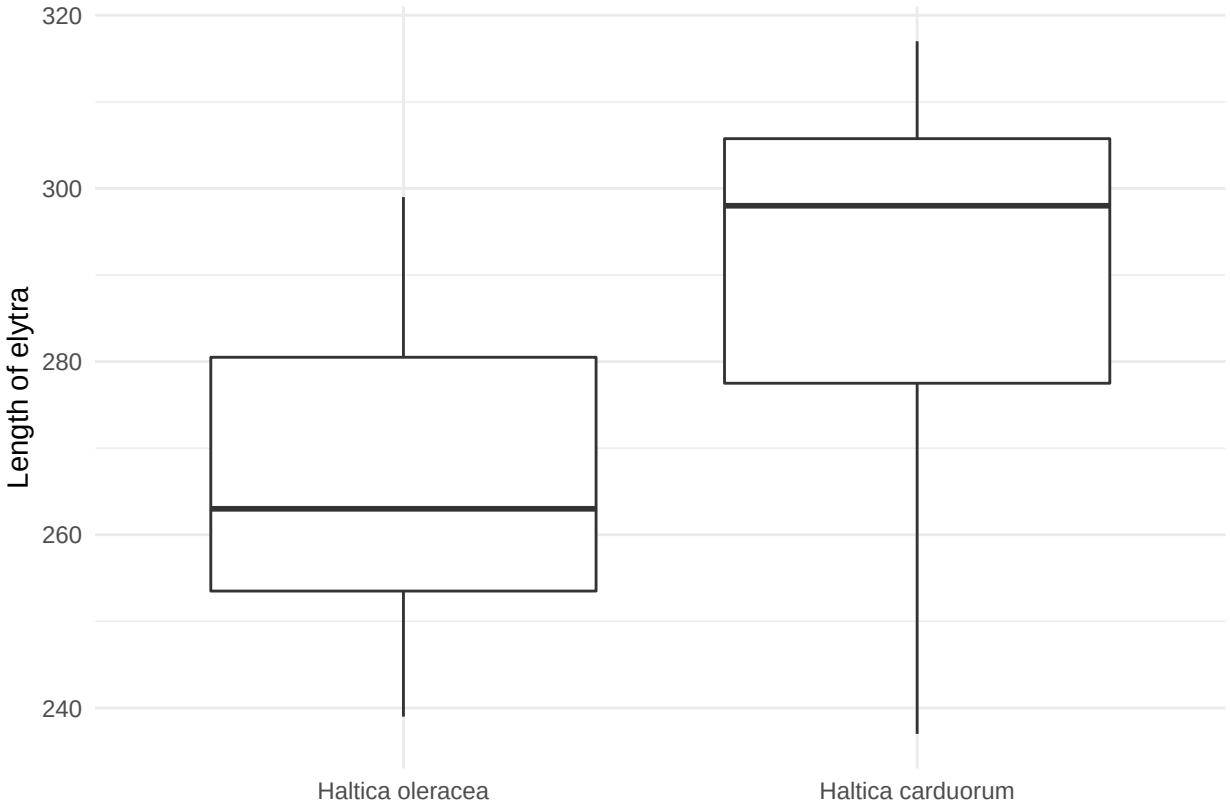
This chart confirms that the data is not normally distributed.

Var1	Var2	Freq	Relationship	Strength
y3	y2	0.73	Positive	Strong
y4	y2	0.59	Positive	Moderate
y4	y3	0.59	Positive	Moderate
y2	y1	0.18	Positive	Weak
y4	y1	-0.07	Negative	Weak
y3	y1	-0.17	Negative	Weak

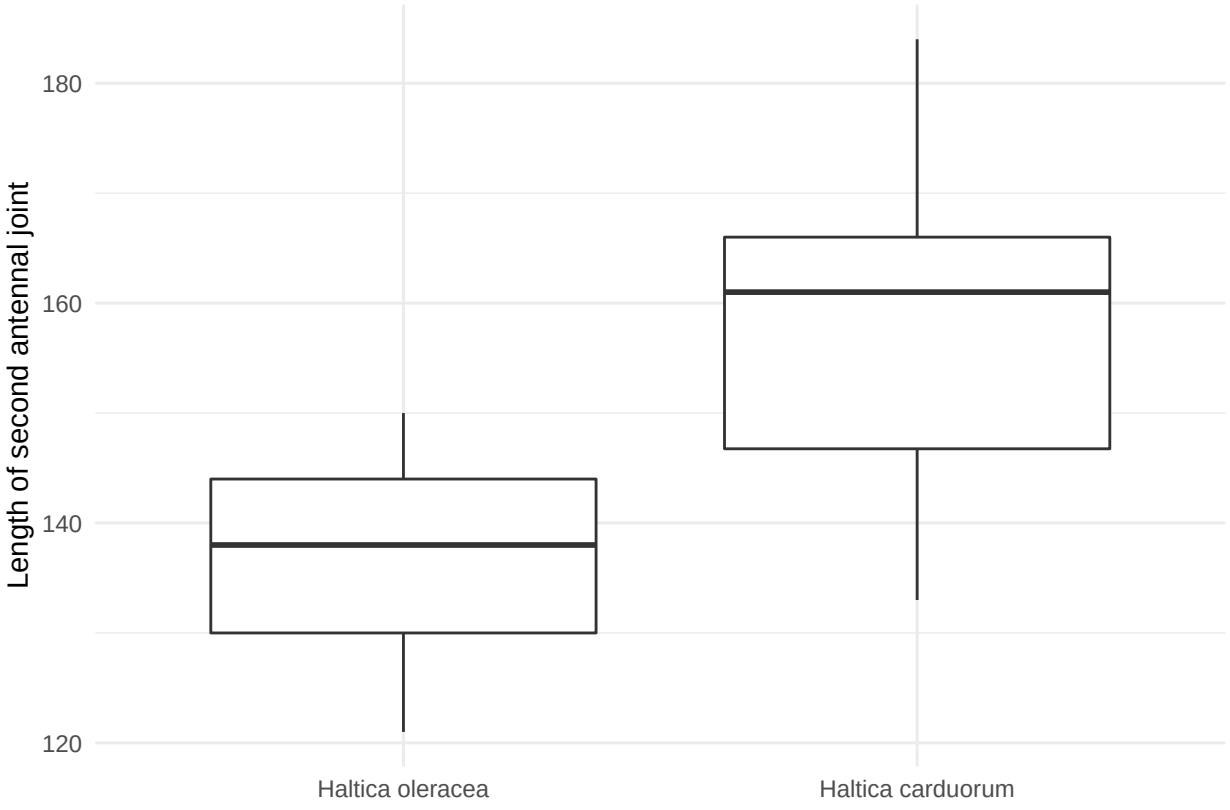
The length of elytra and length of second antennal joint are positively highly correlated. The distance of transverse groove from posterior border of prothorax is weakly correlated with the other 3 variables.



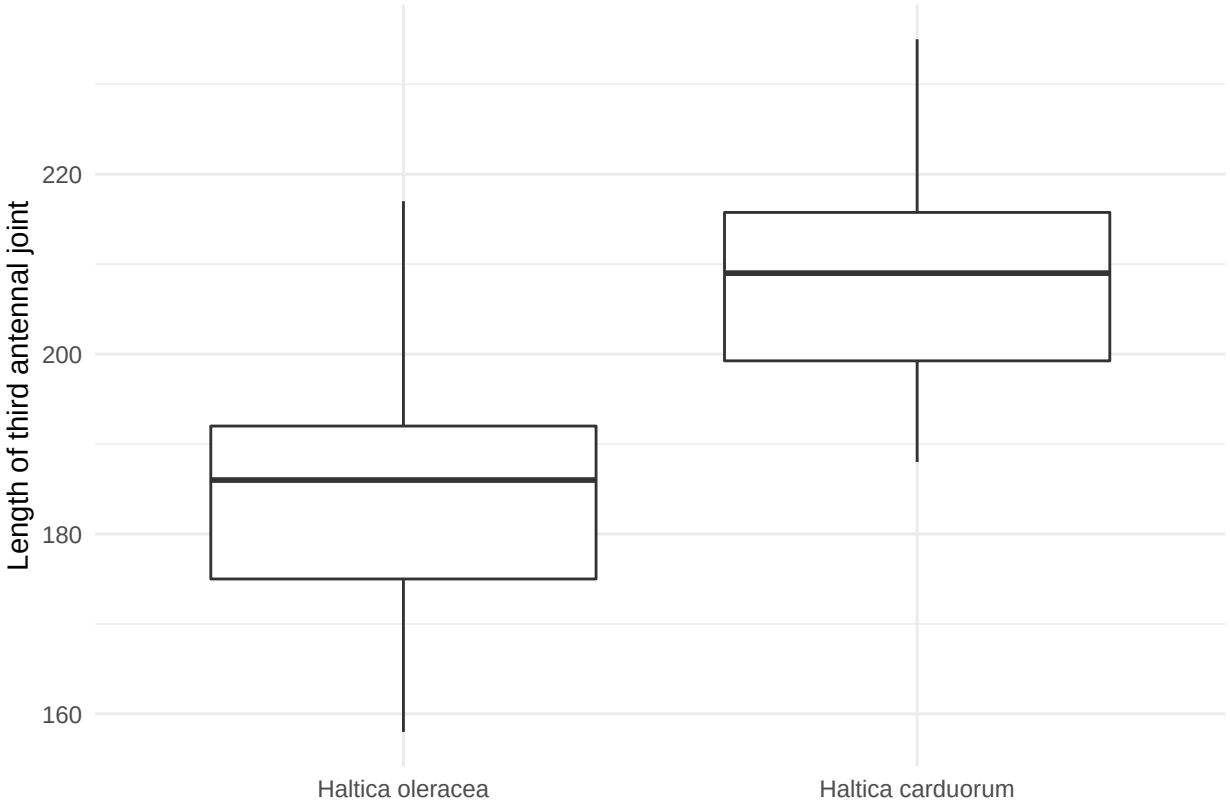
The average distance of transverse groove from posterior border of prothorax appears to be significantly differernt. There are outlier points and data is skewed.



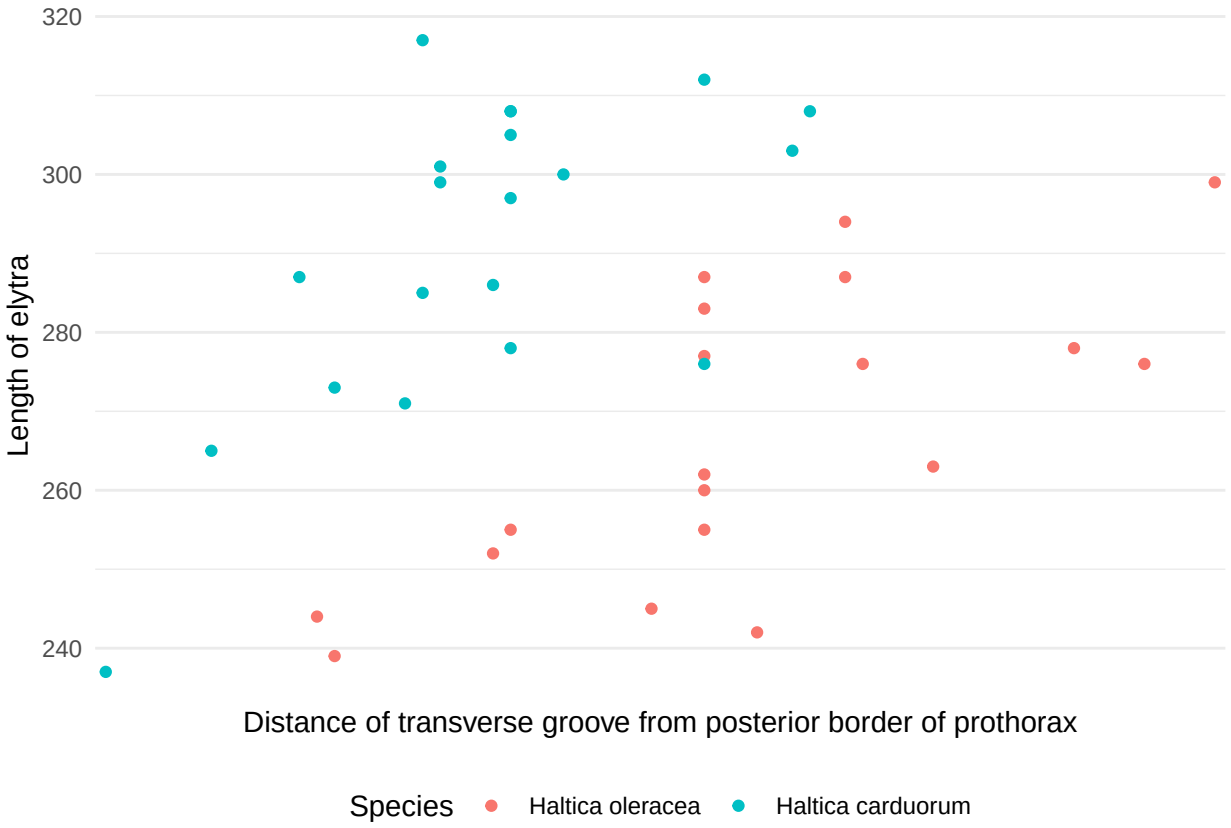
The average distance of transverse groove from posterior border of prothorax appears to be significantly differernt. The data appears to be skewed.



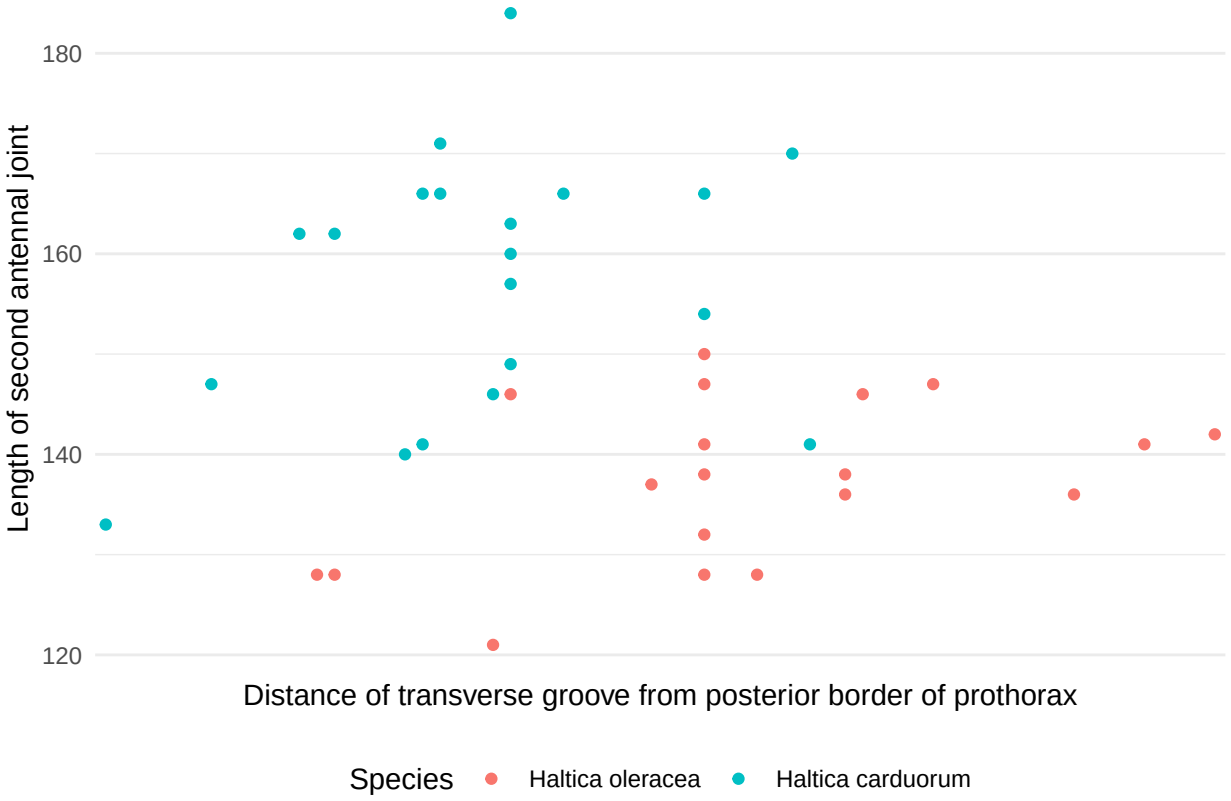
The average length of second antennal joint appear to be significantly differernt. The data appears to be skewed.



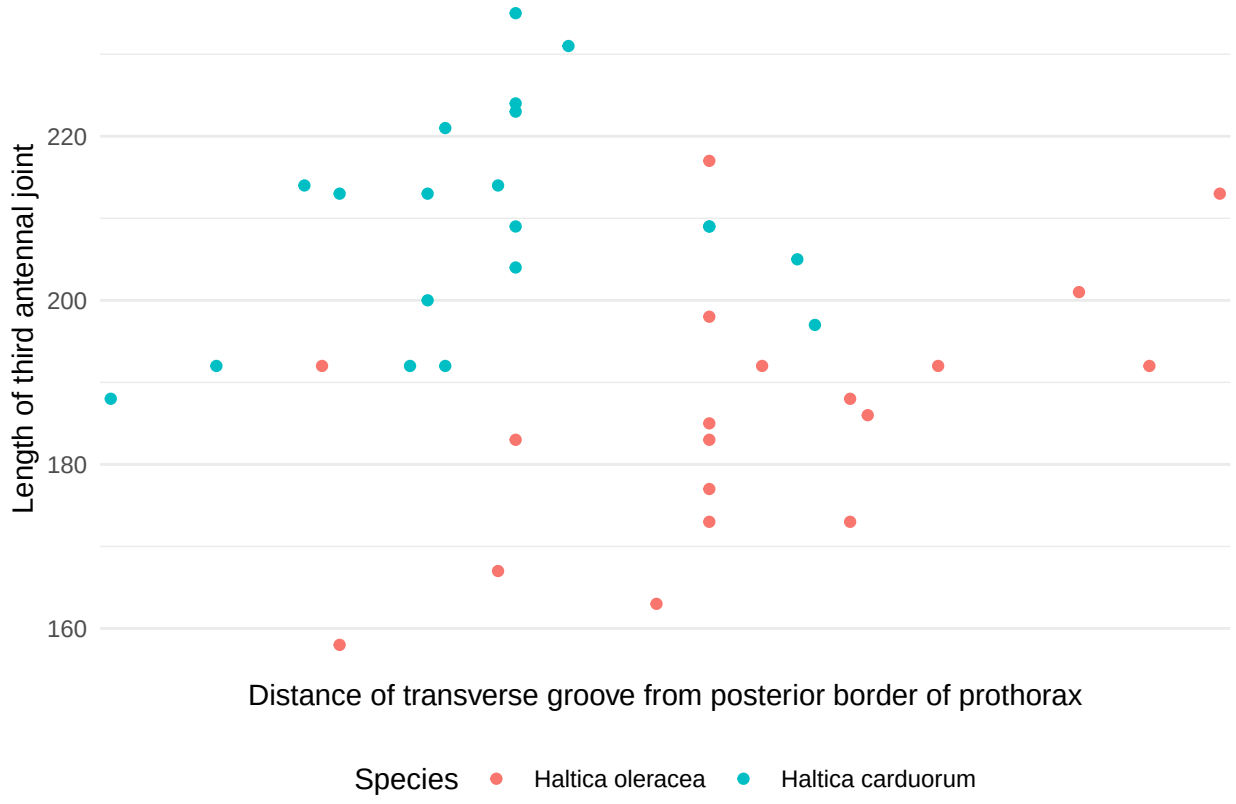
The average length of third antennal joint appear to be significantly differernt. The data appears to be skewed.



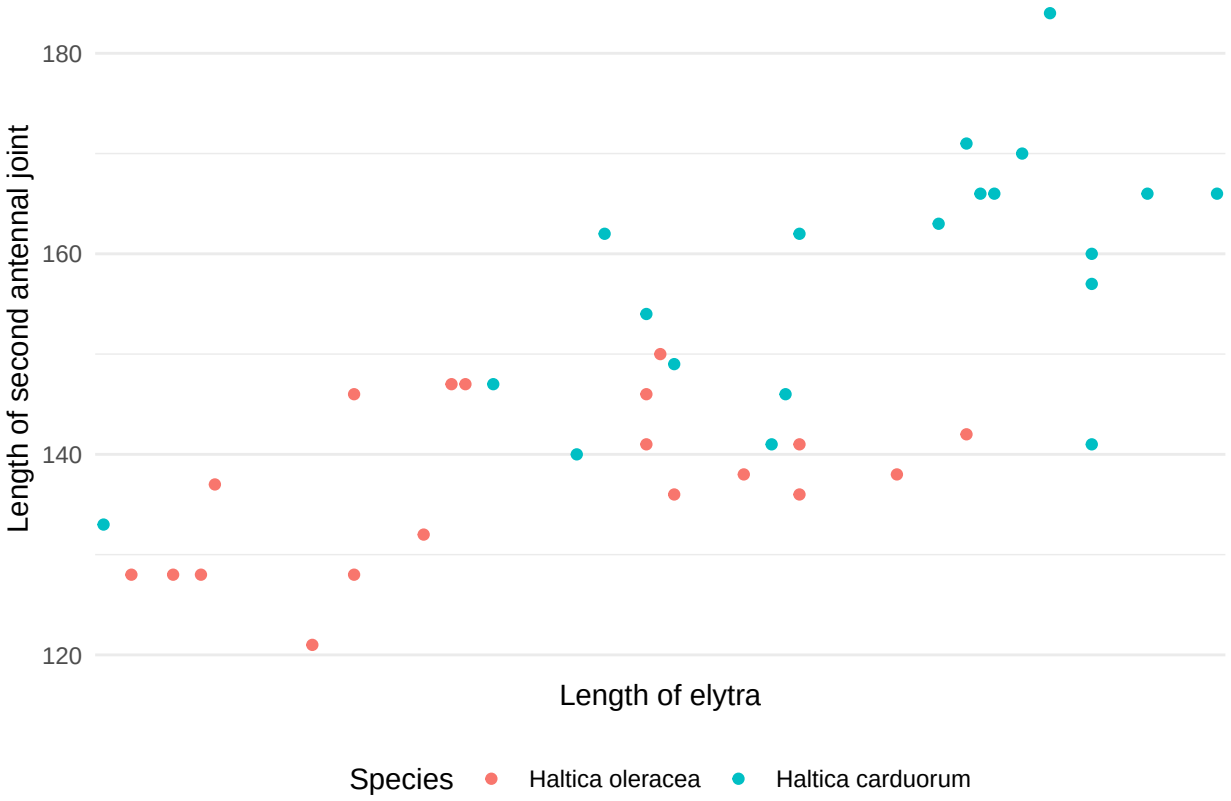
The data points are more spread out over the graph and a few points overlap.



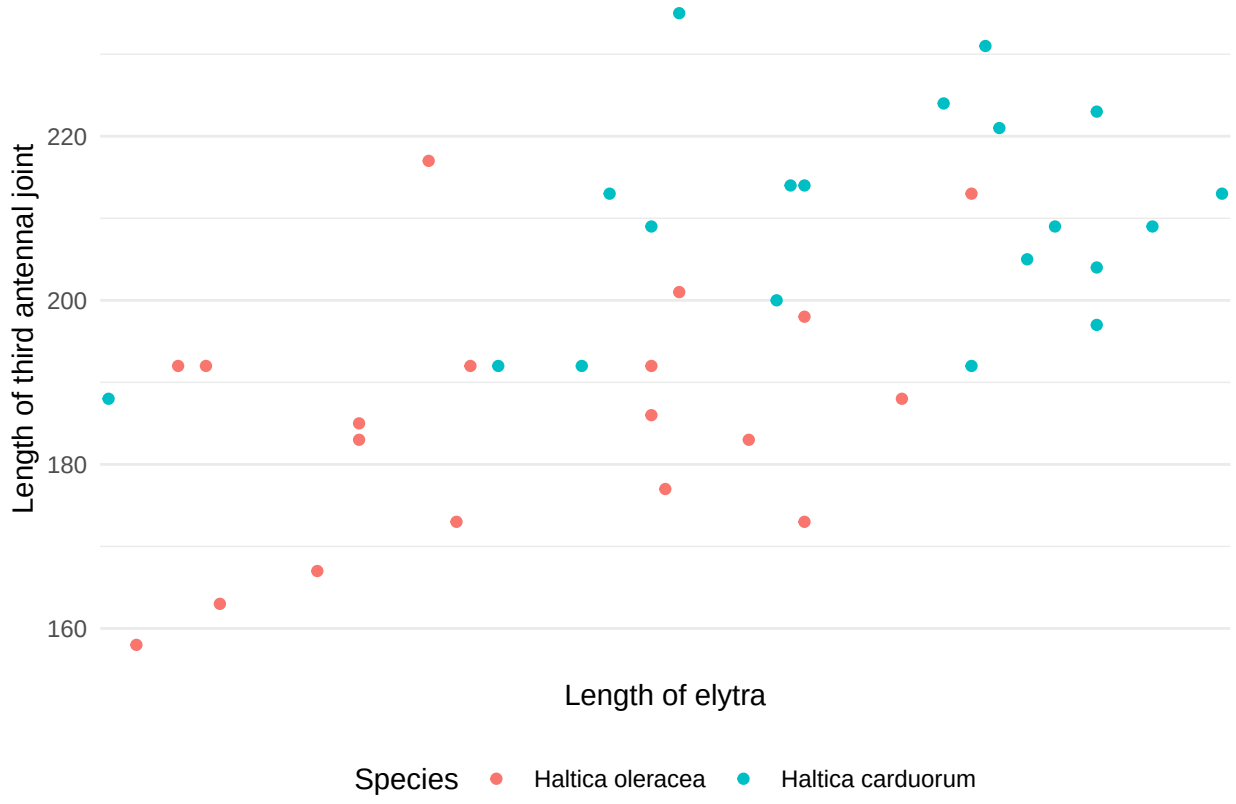
The data points are more spread out over the graph and a few more points overlap than the previous chart.



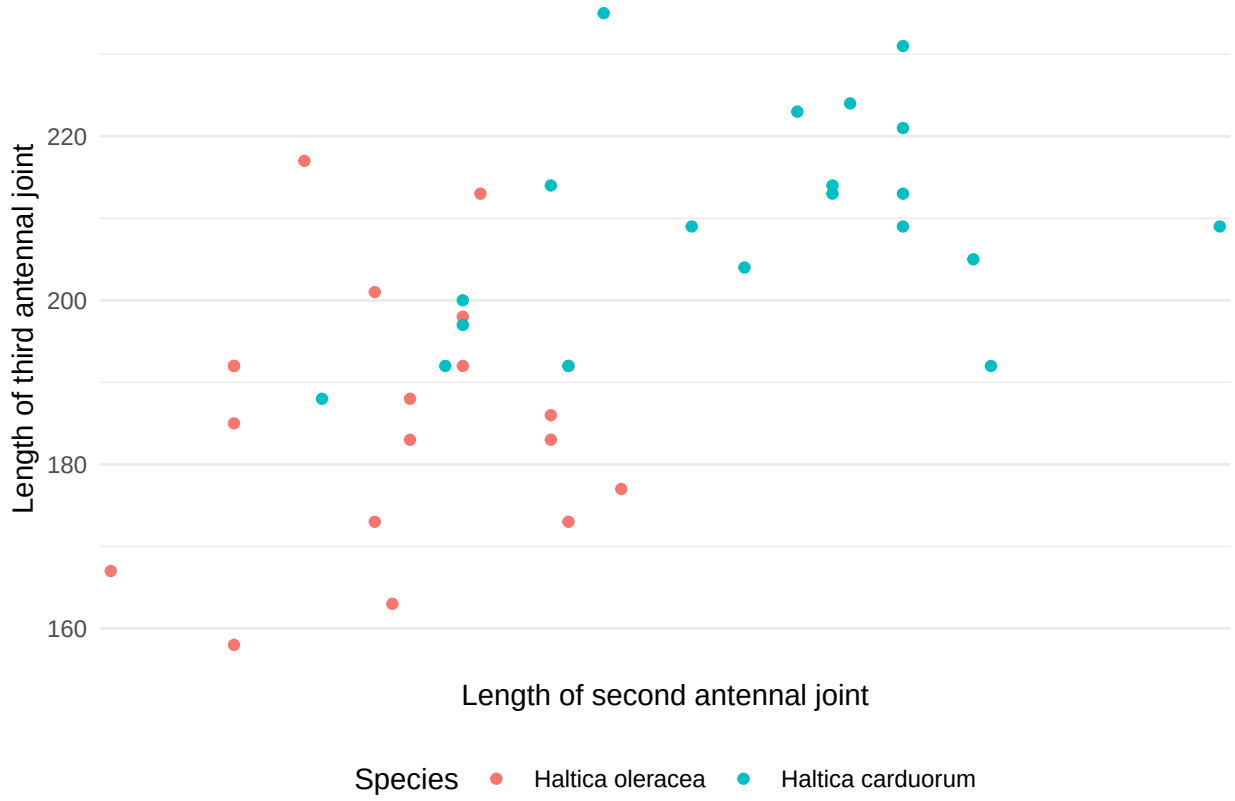
The data points are more spread out over the graph and some points overlap.



The data points are more spread out over the graph and more points overlap than the previous plots.



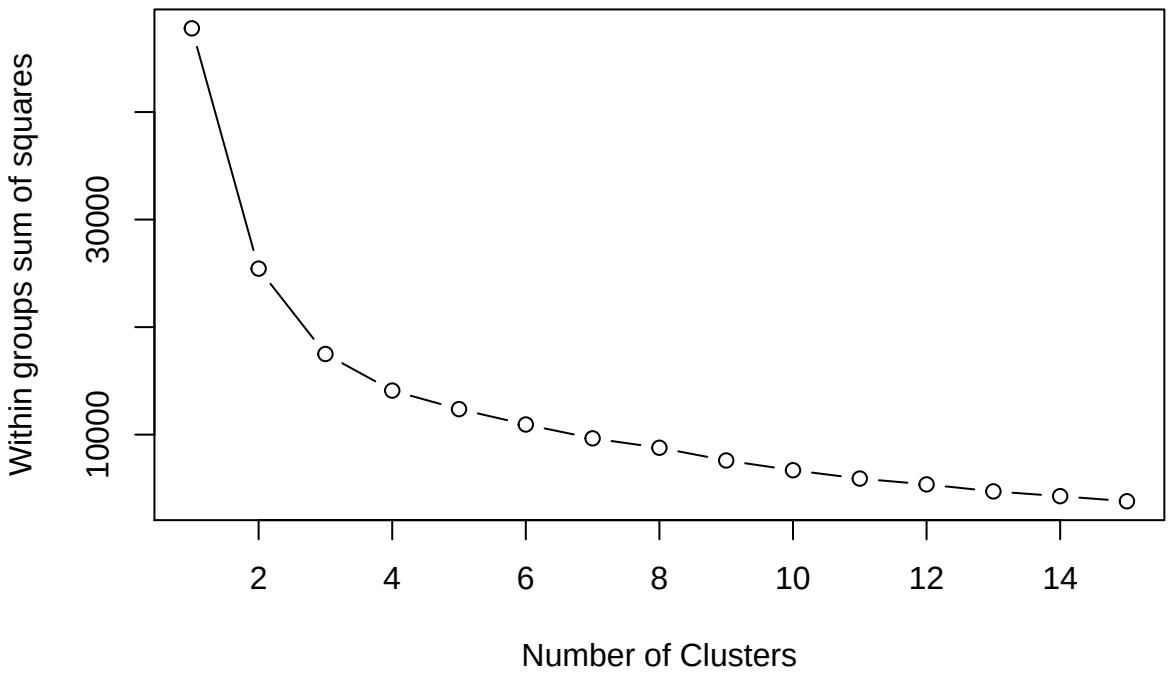
The data points are more spread out over the graph and some points overlap.



The data points are more spread out over the graph and some points overlap.

Computing KNN classifier

This tests will use different possible values of k and fits the final best KNN model that explains the best the beetle data.



The scree plot suggest K = 5.

Find the classification table using the nearest neighbor method.

Nearest neighbor method using caret package

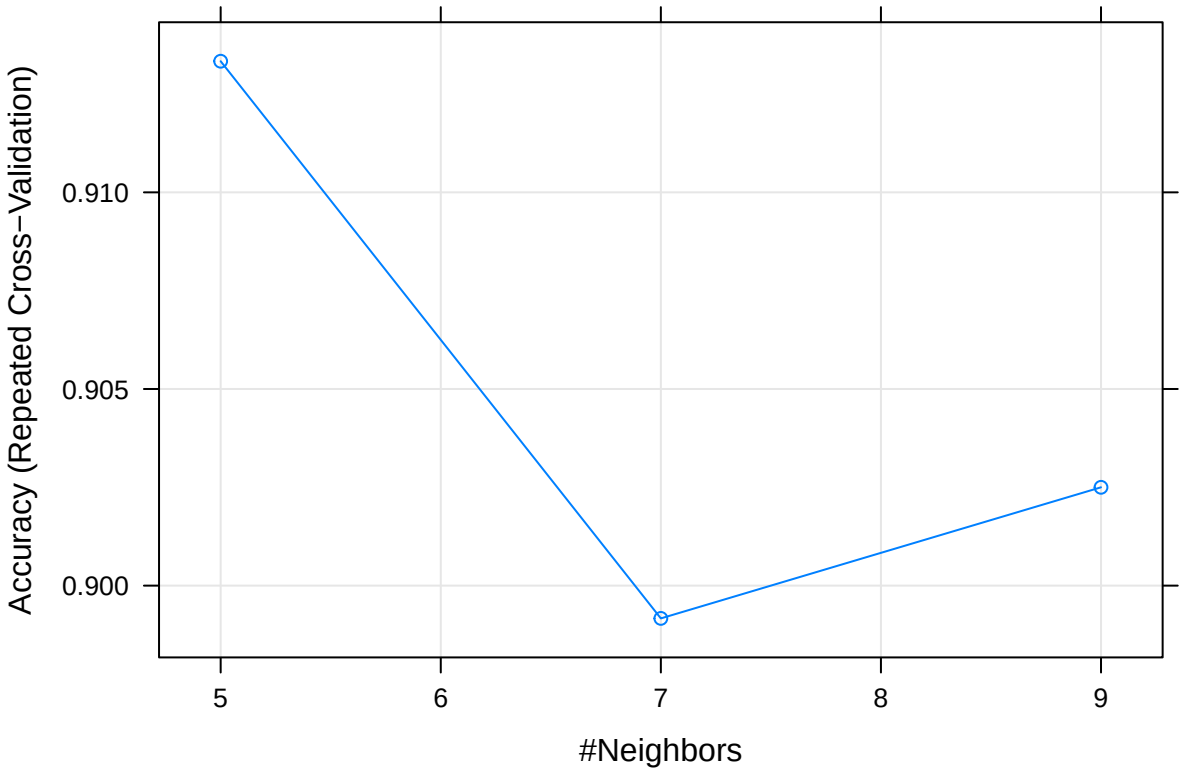
No scaling

```
knnFit <- train(species ~ ., data = data, method = "knn", trControl = ctrl)
knnFit
```

```
## k-Nearest Neighbors
##
## 39 samples
##  4 predictor
##  2 classes: '1', '2'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 35, 35, 35, 35, 36, 35, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.9133333  0.832
##   7  0.8991667  0.802
##   9  0.9025000  0.808
```

```
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

This shows the Accuracy and Kappa metrics result for different k value. From the results, it automatically selects best k-value. Here, our training model is choosing k = 5 as its final value.




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 19  0
##          2  1 19
##
##                Accuracy : 0.9744
##                  95% CI : (0.8652, 0.9994)
##     No Information Rate : 0.5128
##     P-Value [Acc > NIR] : 1.858e-10
##
##                   Kappa : 0.9488
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9500
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9500
```

```
##              Prevalence : 0.5128
##          Detection Rate : 0.4872
##    Detection Prevalence : 0.4872
##       Balanced Accuracy : 0.9750
##
##        'Positive' Class : 1
##
```

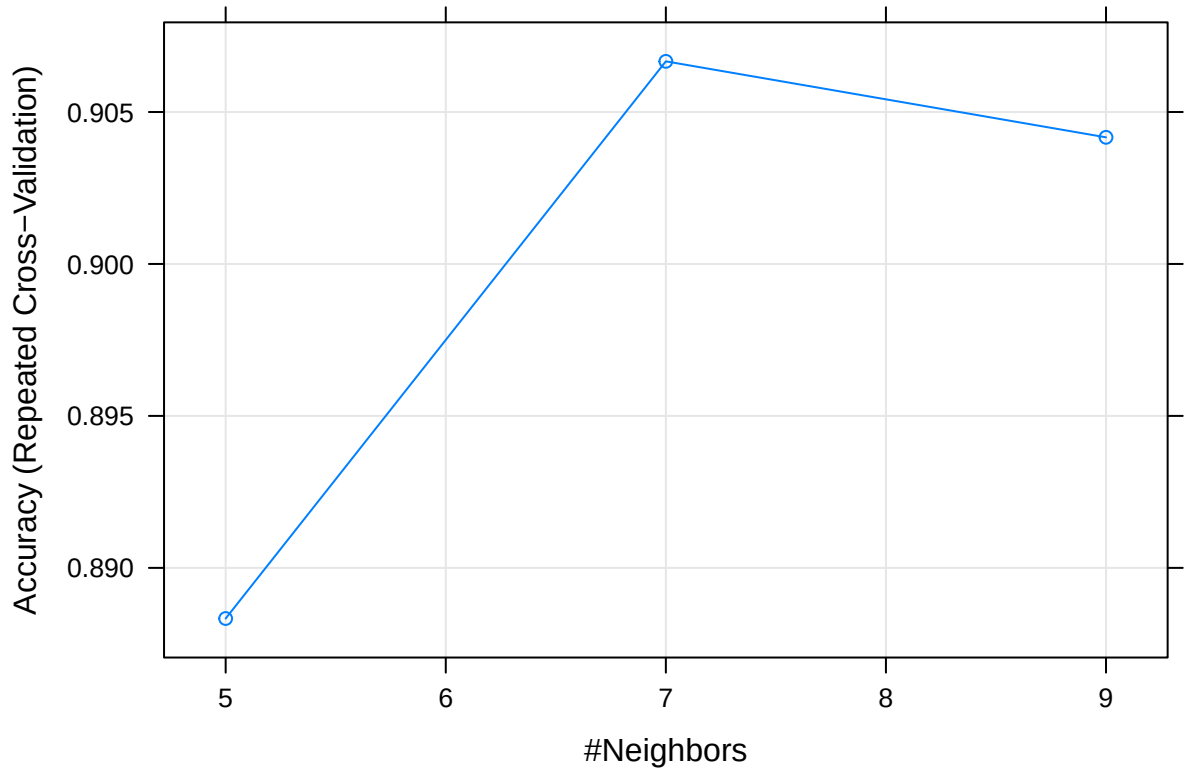
species	y1	y2	y3	y4	prediction
1	189	245	137	163	1
1	192	260	132	217	1
1	217	276	141	192	1
1	221	299	142	213	1
1	171	239	128	158	1
1	192	262	147	173	1
1	213	278	136	201	1
1	192	255	128	185	1
1	170	244	128	192	1
1	201	276	146	186	1
1	195	242	128	192	1
1	205	263	147	192	1
1	180	252	121	167	1
1	192	283	138	183	1
1	200	294	138	188	1
1	192	277	150	177	1
1	200	287	136	173	1
1	181	255	146	183	1
1	192	287	141	198	1
2	181	305	184	209	2
2	158	237	133	188	1
2	184	300	166	231	2
2	171	273	162	213	2
2	181	297	163	224	2
2	181	308	160	223	2
2	177	301	166	221	2
2	198	308	141	197	2
2	180	286	146	214	2
2	177	299	171	192	2
2	176	317	166	213	2
2	192	312	166	209	2
2	176	285	141	200	2
2	169	287	162	214	2
2	164	265	147	192	2
2	181	308	157	204	2
2	192	276	154	209	2
2	181	278	149	235	2
2	175	271	140	192	2
2	197	303	170	205	2

Scaling

```r
knnFit <- train(species ~ ., data = data, method = "knn", trControl = ctrl,
                preProcess = c("center", "scale"))
knnFit
```

```
## k-Nearest Neighbors
##
## 39 samples
##  4 predictor
##  2 classes: '1', '2'
##
## Pre-processing: centered (4), scaled (4)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 35, 35, 35, 35, 35, 35, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.8883333  0.776
##   7  0.9066667  0.816
##   9  0.9041667  0.811
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 7.
```

This shows the Accuracy and Kappa metrics result for different k value. From the results, it automatically selects best k-value. Here, our training model is choosing k = 7 as its final value. The Accuracy is lower than the previous model.



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 19  0
##          2  2 18
##
##                Accuracy : 0.9487
##                  95% CI : (0.8268, 0.9937)
##     No Information Rate : 0.5385
##     P-Value [Acc > NIR] : 1.895e-08
##
##                   Kappa : 0.8976
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 0.9048
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9000
##              Prevalence : 0.5385
##          Detection Rate : 0.4872
##    Detection Prevalence : 0.4872
##       Balanced Accuracy : 0.9524
##
##        'Positive' Class : 1
##
```

species	y1	y2	y3	y4	prediction
1	189	245	137	163	1
1	192	260	132	217	1
1	217	276	141	192	1
1	221	299	142	213	1
1	171	239	128	158	1
1	192	262	147	173	1
1	213	278	136	201	1
1	192	255	128	185	1
1	170	244	128	192	1
1	201	276	146	186	1
1	195	242	128	192	1
1	205	263	147	192	1
1	180	252	121	167	1
1	192	283	138	183	1
1	200	294	138	188	1
1	192	277	150	177	1
1	200	287	136	173	1
1	181	255	146	183	1
1	192	287	141	198	1
2	181	305	184	209	2
2	158	237	133	188	1
2	184	300	166	231	2
2	171	273	162	213	2
2	181	297	163	224	2
2	181	308	160	223	2
2	177	301	166	221	2
2	198	308	141	197	1
2	180	286	146	214	2
2	177	299	171	192	2
2	176	317	166	213	2
2	192	312	166	209	2
2	176	285	141	200	2
2	169	287	162	214	2
2	164	265	147	192	2
2	181	308	157	204	2
2	192	276	154	209	2
2	181	278	149	235	2
2	175	271	140	192	2
2	197	303	170	205	2

Nearest neighbor method using class package

Nearest neighbor method K = 2

```
data$prediction <-  knn(train = train.data, test = train.data, cl = as.matrix(label.data), k = 2)
```

Classification Table for the Beetle Data of Table 5.5 Using the k Nearest Neighbor Method with k=2

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction  1  2
##          1 19  0
##          2  2 18
##
##                Accuracy : 0.9487
##                  95% CI : (0.8268, 0.9937)
##     No Information Rate : 0.5385
##     P-Value [Acc > NIR] : 1.895e-08
##
##                   Kappa : 0.8976
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 0.9048
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9000
##              Prevalence : 0.5385
##          Detection Rate : 0.4872
##    Detection Prevalence : 0.4872
##       Balanced Accuracy : 0.9524
##
##        'Positive' Class : 1
##
```

Correct classification rate = 0.949

Error rate = 0.051

Nearest neighbor method K = 3

```
data$prediction <-  knn(train = train.data, test = train.data, cl = as.matrix(label.data), k = 3)
```

Classification Table for the Beetle Data of Table 5.5 Using the k Nearest Neighbor Method with k=3

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 19  0
##          2  1 19
##
##                Accuracy : 0.9744
##                  95% CI : (0.8652, 0.9994)
##     No Information Rate : 0.5128
##     P-Value [Acc > NIR] : 1.858e-10
##
##                   Kappa : 0.9488
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9500
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
```

```
##          Neg Pred Value : 0.9500
##              Prevalence : 0.5128
##          Detection Rate : 0.4872
##    Detection Prevalence : 0.4872
##       Balanced Accuracy : 0.9750
##
##        'Positive' Class : 1
##
```

Correct classification rate = 0.974

Error rate = 0.026

Nearest neighbor method K = 4

```
data$prediction <-  knn(train = train.data, test = train.data, cl = as.matrix(label.data), k = 4)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 19  0
##          2  0 20
##
##                Accuracy : 1
##                  95% CI : (0.9097, 1)
##     No Information Rate : 0.5128
##     P-Value [Acc > NIR] : 4.883e-12
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.4872
##          Detection Rate : 0.4872
##    Detection Prevalence : 0.4872
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 1
##
```

Correct classification rate = 1

Error rate = 0

Nearest neighbor method K = 5

```
data$prediction <-  knn(train = train.data, test = train.data, cl = as.matrix(label.data), k = 5)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 19  0
##          2  1 19
##
##                Accuracy : 0.9744
##                  95% CI : (0.8652, 0.9994)
##     No Information Rate : 0.5128
##     P-Value [Acc > NIR] : 1.858e-10
##
##                   Kappa : 0.9488
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9500
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9500
##              Prevalence : 0.5128
##          Detection Rate : 0.4872
##    Detection Prevalence : 0.4872
##       Balanced Accuracy : 0.9750
##
##        'Positive' Class : 1
##
```

Correct classification rate = 0.974

Error rate = 0.026

This model predict similarly as the model in part (b). The correct classification rate and error rate are the same as the rates in model in part (b).