Stat 4510/7510 Homework 2

Instructions:
- Please list your name and student number clearly.
- In order to receive credit for a problem, your solution must show sufficient details so that the grader can determine how you obtained your answer.

1.

The dataset `seeds.csv` consists of measurements of geometrical properties of 210 wheat kernels belonging to three different varieties (Kama, Rosa and Canadian). High quality visualization of the internal kernel structure was detected using a soft X-ray technique to construct the following seven, real-valued attributes:

- Area
- Perimeter
- Compactness $C = 4\pi \frac{A}{P^2}$
- Length of kernel
- Width of kernel
- Asymmetry coefficient
- Length of kernel groove

(a). Use the `read.csv()` function to read the data into R. Redefine the “Type” variable from (1,2,3) to (Kama, Rosa, Canadian), using, for example,

```r
seeds$Type[which(seeds$Type==1)] = "Kama"
```

In addition, ensure that R knows that this variable is a factor with the line

```r
seeds$Type = factor(seeds$Type)
```

(b). Produce a summary table of the dataset.
```
summary(seeds)
```

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>10.59</td>
<td>12.27</td>
<td>14.36</td>
<td>14.85</td>
<td>17.30</td>
<td>21.18</td>
</tr>
<tr>
<td>perimeter</td>
<td>12.41</td>
<td>13.45</td>
<td>14.32</td>
<td>14.56</td>
<td>15.71</td>
<td>17.25</td>
</tr>
<tr>
<td>compactness</td>
<td>0.8081</td>
<td>0.8569</td>
<td>0.8734</td>
<td>0.8710</td>
<td>0.8878</td>
<td>0.9183</td>
</tr>
<tr>
<td>length.of.kernel</td>
<td>4.899</td>
<td>5.262</td>
<td>5.524</td>
<td>5.629</td>
<td>5.980</td>
<td>6.675</td>
</tr>
<tr>
<td>width.of.kernel</td>
<td>2.630</td>
<td>2.944</td>
<td>3.237</td>
<td>3.259</td>
<td>3.562</td>
<td>4.033</td>
</tr>
<tr>
<td>asymmetry.coefficient</td>
<td>0.7651</td>
<td>2.5615</td>
<td>3.5990</td>
<td>3.7002</td>
<td>4.7687</td>
<td>8.4560</td>
</tr>
<tr>
<td>length.of.kernel.groove</td>
<td>4.519</td>
<td>5.045</td>
<td>5.223</td>
<td>5.408</td>
<td>5.877</td>
<td>6.550</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canadian</td>
<td>70</td>
</tr>
<tr>
<td>Kama</td>
<td>70</td>
</tr>
<tr>
<td>Rosa</td>
<td>70</td>
</tr>
</tbody>
</table>

(c). What proportion of observations have perimeter values greater than 15?

```
length(which(seeds$perimeter > 15))/length(seeds$perimeter)
```

## [1] 0.3380952

(d). Which observation has the largest Asymmetry coefficient? To which class does this observation belong?

```
max(seeds$asymmetry.coefficient)
seeds[which(seeds$asymmetry.coefficient == max(seeds$asymmetry.coefficient)),]
```

The Canadian kernel type has the largest asymmetry coefficient.

(e). Produce a scatterplot matrix of all variables and note some relationships between them. Which attributes are highly related? Which attributes do a good job of distinguishing type?

```
pairs(seeds)
```
The highly correlated attributes are: area, perimeter, length.of.kernel, width.of.kernel, and length.of.kernel.groove. Attributes that show distinctions in kernel type include area, perimeter, length.of.kernel, and width.of.kernel.

(f). Select two or three attributes and produce boxplots of these attributes vs. Type (don't forget axis labels!). What two or three variables might you would want to use as predictor variables in a model for kernel type based on these plots. Why?

```R
boxplot(seeds$Area~seeds$Type, xlab="Kernel type", ylab="Kernel area")
```
```
boxplot(seeds$compactness~seeds$Type, 
xlab="Kernel type", ylab="Kernel compactness")
```
boxplot(seeds$length.of.kernel.groove~seeds$Type, xlab="Kernel type", ylab="Kernel groove length")
I'd suggest area, compactness, and length.of.kernel.groove. Each of these appears less correlated with each other and shows from distinction in the kernel type.

(g). Augment your pairs() function from part (e) to give a different color to each type. *Hint: col=“green” would make all the points green, so col = [variable] will color them according to a variable.* What additional information does this figure give you in building a model for seed type?

```r
pairs(seeds,col=seeds$Type)
```
This figure helps to differentiate the different types of kernels for each pair of covariates. It offers a nice visualization for determine which covariates might be important in using in a model.

2. 7510*

One approach for classifying variables (e.g., identifying the type of kernel) uses the k nearest neighbors (KNN). In general, the classifications are based on similarities in covariates (we will learn more about this approach in future chapters). We will apply the KNN method to the seeds dataset using the knn() function in the class package. In R, a “package” is a collection of functions which extend the capabilities of your R installation. There are many packages available and we will learn and use many throughout this class. To install a specific package, enter the following code or click the packages tab in Rstudio and select ‘install.’
install.packages("class")
library(class)

(a). First, randomly split the data into a training and test set. We will “holdout” 30% of the
data randomly, and train our KNN model on the remaining 70%. To accomplish this we can
randomly select rows using the sample() function, then define our test dataset as seeds.test
and our training dataset as seeds.train. Be sure to set a seed of 1 for consistent results.

```r
set.seed(1)
TestSamples=sample(1:nrow(seeds),size=.30*nrow(seeds))
seeds.test=seeds[TestSamples,]
seeds.train=seeds[-TestSamples,]
```

Be sure to make seeds.train which consists of the remaining observations

(b). Predict the classes of the test set using the knn() function using K=10. The knn functions
takes the following arguments:

```
type.pred=knn(seeds.train[,1:7],seeds.test[,1:7],seeds.train[,8],k=10)
```

Where train is a matrix of the training attribute variables (no labels!), test is a matrix of
the test attribute variables, cl is a vector with the class labels in the training set, and k is
the number of nearest neighbors.

(c). Find the misclassification rate of your predictions

```
length(which(type.pred!=seeds.test[,8]))/length(type.pred)
```

The misclassification rate is 6.35%.

(d). Investigate the effects of the number of nearest neighbors (k) and the % training vs % testing
on misclassification rate. Comment on your results.

The more training data, the lower the misclassification rate for the test data. As for the num-
ber of neighbors, the misclassification rate will increase for too few and too many neighbors.