1.- Randomly split the dataset into 75% train and 25% test. Note: It can be done by using the sample function to generate integers belonging to the interval [1..size(data set)]. Use these numbers to identify the instances.

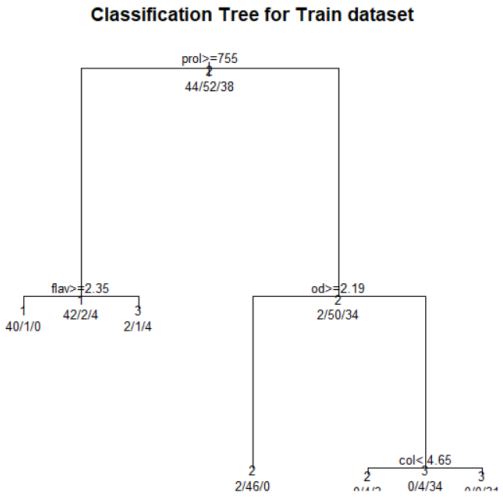
```
> total_rows <- nrow(wine)
> train_size <- round(0.75 * total_rows)
> test_size <- total_rows - train_size
> train_indices <- sample(1:total_rows, train_size)
> train_data <- wine[train_indices, ]
> test_data <- wine[-train_indices, ]
```

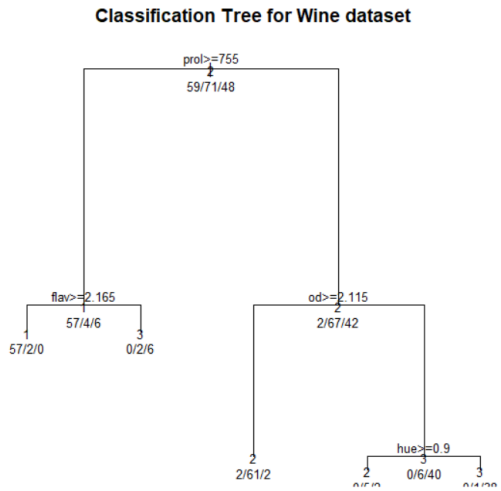
2.- Learn a decision tree using the training set.

model_t<-rpart(class~.,data=train_data,method="class")

3.- Visualise the tree and display the results. Is there any difference with respect to the model trained with the whole dataset?

The prediction model has changed, especially in the lower's branches, instead of evaluating the column "hue" to determine if a tree is of one type or another in the trained dataset we use the column "col".





4.- Use the model to predict the class label for the test set by using the "predict" function. Repeat the predictions but now using the parameter type="class" (use a different variable to keep the new results). What are the differences?

predictions_def <- predict(model_t, newdata = test_data)

```
> predictions_def
                1          2         3
1   0.97560976 0.02439024 0.0000000
2   0.97560976 0.02439024 0.0000000
9   0.97560976 0.02439024 0.0000000
15  0.97560976 0.02439024 0.0000000
20  0.97560976 0.02439024 0.0000000
21  0.97560976 0.02439024 0.0000000
23  0.97560976 0.02439024 0.0000000
29  0.97560976 0.02439024 0.0000000
33  0.97560976 0.02439024 0.0000000
```

predictions_tclass <- predict(model_t, newdata = test_data, type = "class")

```
> predictions_tclass
  1   2   9  15  20  21  23  29  33  34  43  51  52  55  58  62  70  71  75  84  86  87  93
  1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   3   2   3   3   3   2   2   2
 95 103 104 106 107 113 114 115 119 121 123 132 137 141 149 155 167 170 171 176 177
  2   2   2   2   2   2   2   2   2   2   2   3   2   2   3   3   3   3   3   3   3
Levels: 1 2 3
```

The difference between the two of the is that the first one returns a table with the probability of each tree belonging to a specific class. Using type class will create a table where each colum will have the class predicted by the model for that tree.

5.- Calculate the performance of the model when it is applied to the test set by displaying a table that shows the predicted classes versus the real classes

```
> accuracy <- mean(predictions_tclass == test_data$class)
> accuracy
[1] 0.8636364
```

The accuracy of the predicted model applied to the test set is 86,63%

6.- Try some other methods or parameters of the rpart package to see whether you can still improve the results further. You can also compare with other packages.

Using Packages randomForest. I obtained a 93,18% accuracy.

```
> library("randomForest")
> model_try <- randomForest(class~., data = train_data, ntree = 100)
> predictions_try <- predict(model_try, newdata = test_data)
> accuracy <- mean(predictions_try == test_data$class)
> accuracy
[1] 0.9318182
```

Using the libary party, and applying the ctree model we can obtain a 88,6% accuracy

```
> library(party)
> decision_tree_model <- ctree(class~., data = train_data)
> predictions_try <- predict(decision_tree_model, newdata = test_data)
> accuracy <- mean(predictions_try == test_data$class)
> accuracy
[1] 0.8863636
```