To consider

This notebook is configured to execute R instructions directly (you can download the notebook and open it in a normal text editor to check the kernel specification in the metadata).

In your case, the data file must be loaded every time you want to analyze it, since the files disappear when you close the session.

Aim

Construct a classification model that predicts the species to which a flower belongs based on the length and width of the petal and sepal.

There are four predictor variables (x1 = Sepal.Length, x2 = Sepal.Width, x3 = Petal.Length, x4 = Petal.Width) and a class variable (y = Species). The purpose is to find a function f (x1, x2, x3, x4) that predicts the value of y. That is, a function such that y = f (x1, x2, x3, x4) is searched for.

The classification tree is a representation of this function.

Required packages

- **rpart:** contains algorithm implementations to obtain classification trees
- **rpart.plot:** contains functions to represent classification trees
- **caret -** Contains functions for training and drawing classification and regression models. We will use, among others, the functions *confusionMatrix* , to obtain the confusion matrix, and *train* , to fit models of classification trees
- **datasets -** Contains the dataset that will be used during the analysis. Use *library (help = "datasets")* to list available databases
- **dplyr:** contains functions for data manipulation such as row filtering, column selection, row reordering ... We will use the *sample_frac* function to randomly divide the data set

Installing packages and uploading data

We install the necessary packages to obtain the classification tree, represent it and analyze it. We also install the package that contains the data file that we will use to learn the model. Remember that it is not enough to install the package, you must also load it into memory so that you can use the functions included in it. For this we will use the command *library ()*

In [18]:
```
# install.packages ("rpart")
# install.packages ("rpart.plot")
# install.packages ("caret")
# install.packages ("datasets")
# install.packages ("dplyr")
#install. packages ("e1071")
```

```
library ( rpart )
library ( rpart.plot )
library ( caret )
library ( datasets )
library ( dplyr )
library ( e1071 )
```

```
Loading required package: lattice


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
```

Data loading, preliminary analysis and preparation

We load the *iris* data file and view its content. It contains the length and width of the sepal and petal of 150 flowers, along with the species to which they belong. We also show descriptive statistics for each variable and use the *complete.cases ()* function to identify null values.

We observe that the classes are balanced (there are 50 cases of each species) and that there are no null values. Initially, the data is ready for analysis without the need for specific preparation tasks.

In [13]:
```
data ( iris )
```

In [14]:
```
iris
```

A data.frame: 150 × 5

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
<dbl>	<dbl>	<dbl>	<dbl>	<fct>
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
<dbl>	<dbl>	<dbl>	<dbl>	<fct>
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa
5.4	3.4	1.7	0.2	setosa
5.1	3.7	1.5	0.4	setosa
4.6	3.6	1.0	0.2	setosa
5.1	3.3	1.7	0.5	setosa
4.8	3.4	1.9	0.2	setosa
5.0	3.0	1.6	0.2	setosa
5.0	3.4	1.6	0.4	setosa
5.2	3.5	1.5	0.2	setosa
5.2	3.4	1.4	0.2	setosa
4.7	3.2	1.6	0.2	setosa
⋮	⋮	⋮	⋮	⋮
6.9	3.2	5.7	2.3	virginica
5.6	2.8	4.9	2.0	virginica
7.7	2.8	6.7	2.0	virginica
6.3	2.7	4.9	1.8	virginica
6.7	3.3	5.7	2.1	virginica

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
<dbl>	<dbl>	<dbl>	<dbl>	<fct>
7.2	3.2	6.0	1.8	virginica
6.2	2.8	4.8	1.8	virginica
6.1	3.0	4.9	1.8	virginica
6.4	2.8	5.6	2.1	virginica
7.2	3.0	5.8	1.6	virginica
7.4	2.8	6.1	1.9	virginica
7.9	3.8	6.4	2.0	virginica
6.4	2.8	5.6	2.2	virginica
6.3	2.8	5.1	1.5	virginica
6.1	2.6	5.6	1.4	virginica
7.7	3.0	6.1	2.3	virginica
6.3	3.4	5.6	2.4	virginica
6.4	3.1	5.5	1.8	virginica
6.0	3.0	4.8	1.8	virginica
6.9	3.1	5.4	2.1	virginica
6.7	3.1	5.6	2.4	virginica
6.9	3.1	5.1	2.3	virginica
5.8	2.7	5.1	1.9	virginica
6.8	3.2	5.9	2.3	virginica
6.7	3.3	5.7	2.5	virginica
6.7	3.0	5.2	2.3	virginica
6.3	2.5	5.0	1.9	virginica
6.5	3.0	5.2	2.0	virginica
6.2	3.4	5.4	2.3	virginica
5.9	3.0	5.1	1.8	virginica

In [15]:
```
summary ( iris )
```

```
 Sepal.Length Sepal.Width Petal.Length Petal.Width
Min.: 4,300 Min.: 2,000 Min.: 1,000 Min.: 0.100
1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300
Median: 5,800 Median: 3,000 Median: 4,350 Median: 1,300
Mean: 5.843 Mean: 3.057 Mean: 3.758 Mean: 1.199
3rd Qu.:6,400 3rd Qu.:3,300 3rd Qu.:5,100 3rd Qu.:1,800
```

```
Max. : 7,900 Max. : 4,400 Max. : 6,900 Max. : 2,500
      Species
setosa: 50
versicolor: 50
virginica: 50
```

In [16]:
```
iris [ ! complete.cases ( iris ),]
```

A data.frame: 0 × 5

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
<dbl>	<dbl>	<dbl>	<dbl>	<fct>

Obtaining a classification model

We obtain a first classification tree that predicts the species using the length and width of the sepal as predictor variables. We use the *rpart* function for this . In addition, we visualize the tree obtained

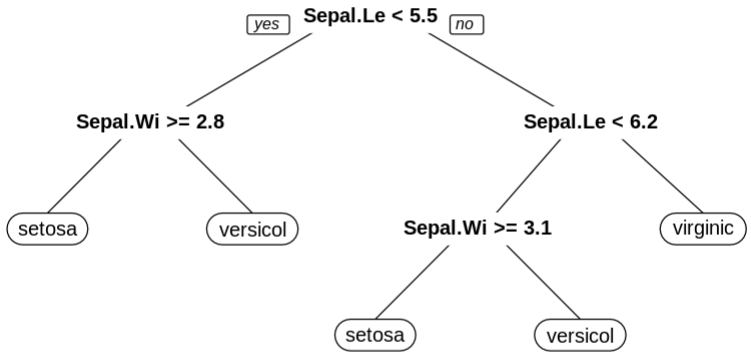
In [26]:
```
iris.tree1  <-  rpart ( Species  ~  Sepal.Length  +  Sepal.Width ,  iris ,  method  =  "c
```

In [27]:
```
iris.tree1
```

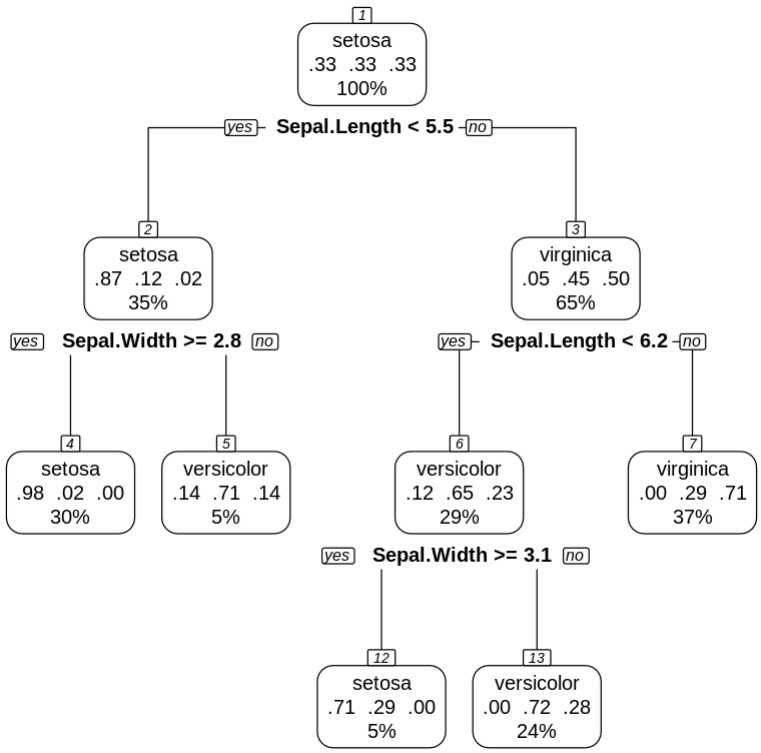
```
n = 150

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 150 100 setosa (0.33333333 0.33333333 0.33333333)
   2) Sepal.Length <5.45 52 7 setosa (0.86538462 0.11538462 0.01923077)
     4) Sepal.Width> = 2.8 45 1 setosa (0.97777778 0.02222222 0.00000000) *
     5) Sepal.Width <2.8 7 2 versicolor (0.14285714 0.71428571 0.14285714) *
   3) Sepal.Length> = 5.45 98 49 virginica (0.05102041 0.44897959 0.50000000)
     6) Sepal.Length <6.15 43 15 versicolor (0.11627907 0.65116279 0.23255814)
      12) Sepal.Width> = 3.1 7 2 setosa (0.71428571 0.28571429 0.00000000) *
      13) Sepal.Width <3.1 36 10 versicolor (0.00000000 0.72222222 0.27777778) *
     7) Sepal.Length> = 6.15 55 16 virginica (0.00000000 0.29090909 0.70909091) *
```

Now we draw the classification tree using the *prp* function of the *rpart.plot* package

In [28]:
```
prp ( iris.tree1 )
```



In [29]:
```
prp ( iris.tree1 ,  type  =  2 ,  extra  = "auto" ,  nn  =  TRUE ,  branch  =  1 ,  varle
```



In [30]:
```
predicted1  <-  predict ( iris.tree1 ,  newdata  =  iris ,  type  =  "class" )
confusionMatrix ( predicted1 ,  iris [[ "Species" ]])
```

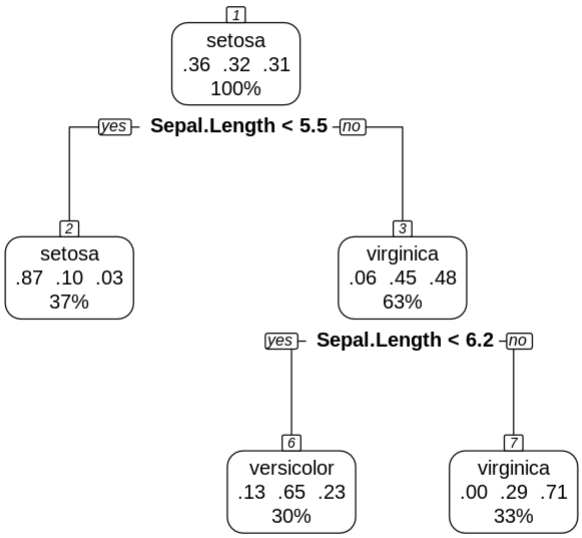
```
Confusion Matrix and Statistics

           Reference
Prediction setosa versicolor virginica
  setosa 49 3 0
  versicolor 1 31 11
  virginica 0 16 39

Overall Statistics

               Accuracy: 0.7933
                 95% CI: (0.7197, 0.8551)
    No Information Rate: 0.3333
    P-Value [Acc> NIR]: <2.2e-16

                  Kappa: 0.69

 Mcnemar's Test P-Value: NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity 0.9800 0.6200 0.7800
Specificity 0.9700 0.8800 0.8400
Pos Pred Value 0.9423 0.7209 0.7091
Neg Pred Value 0.9898 0.8224 0.8842
Prevalence 0.3333 0.3333 0.3333
Detection Rate 0.3267 0.2067 0.2600
Detection Prevalence 0.3467 0.2867 0.3667
Balanced Accuracy 0.9750 0.7500 0.8100
```

We obtain the classification tree again, but considering a different training set from the validation set. To do this, we divide the data set into two sets: one for the training or adjustment of the model and the other for its validation. Also, we get the confusion matrix and some statistics associated with the classifier.

In [31]:
```
iris_training <- sample_frac ( iris , 0.7 )
iris_validation <- setdiff ( iris , iris_training )
```

In [32]:
```
iris.tree2 <- rpart ( Species ~ Sepal.Length + Sepal.Width , iris_training , meth
```

In [33]:
```
prp ( iris.tree2 , type = 2 , extra = "auto" , nn = TRUE , branch = 1 , varle
```



In [34]:
```
predicted2 <- predict ( iris.tree2 , newdata = iris_validation , type = "class" )
confusionMatrix ( predicted2 , iris_validation [[ "Species" ]])
```

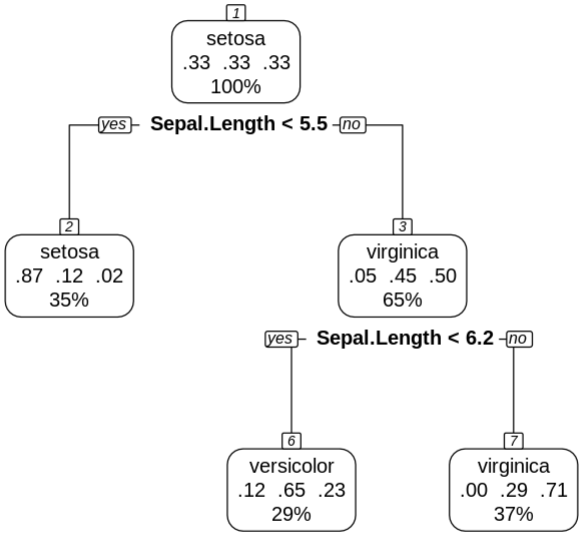
```
Confusion Matrix and Statistics

            Reference
Prediction setosa versicolor virginica
  setosa 11 2 0
  versicolor 1 8 2
  virginica 0 6 14

Overall Statistics

               Accuracy: 0.75
                 95% CI: (0.5966, 0.8681)
    No Information Rate: 0.3636
    P-Value [Acc> NIR]: 2.06e-07

                  Kappa: 0.6231

 Mcnemar's Test P-Value: NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity 0.9167 0.5000 0.8750
Specificity 0.9375 0.8929 0.7857
Pos Pred Value 0.8462 0.7273 0.7000
Neg Pred Value 0.9677 0.7576 0.9167
Prevalence 0.2727 0.3636 0.3636
Detection Rate 0.2500 0.1818 0.3182
Detection Prevalence 0.2955 0.2500 0.4545
Balanced Accuracy 0.9271 0.6964 0.8304
```

Finally, we apply cross-validation to obtain and validate the classification tree. We use the *train ()*

function of the *caret* package for this . We must specify the variable to predict and the predictor variables, the data set (*data*), the method to obtain the model (*rpart*) and the parameters that control the execution of the *train ()* function . In this case, we indicate that cross validation (*method = "cv"*) will be used with 10 folds (*number = 10*).

In [35]:
```
train.control <- trainControl ( method = "cv" , number = 10 )
iris.tree3 <- train ( Species ~ Sepal.Length + Sepal.Width , data = iris , met
prp ( iris.tree3 $ finalModel , type = 2 , extra = "auto" , nn = TRUE , branch =

predicted3 <- predict ( iris.tree3 $ finalModel , newdata = iris_validation , type
confusionMatrix ( predicted3 , iris_validation [[ "Species" ]])
```

```
Confusion Matrix and Statistics

            Reference
Prediction setosa versicolor virginica
  setosa 11 2 0
  versicolor 1 8 2
  virginica 0 6 14

Overall Statistics

               Accuracy: 0.75
                 95% CI: (0.5966, 0.8681)
    No Information Rate: 0.3636
    P-Value [Acc> NIR]: 2.06e-07

                  Kappa: 0.6231

 Mcnemar's Test P-Value: NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity 0.9167 0.5000 0.8750
Specificity 0.9375 0.8929 0.7857
Pos Pred Value 0.8462 0.7273 0.7000
Neg Pred Value 0.9677 0.7576 0.9167
Prevalence 0.2727 0.3636 0.3636
Detection Rate 0.2500 0.1818 0.3182
Detection Prevalence 0.2955 0.2500 0.4545
Balanced Accuracy 0.9271 0.6964 0.8304
```



TASK: Individual practice

Statement

1. Represent on a graph the information regarding the length (x) and width (y) of the flower petals, identifying the species of each flower with a different color.
2. Obtain a classification tree to predict the species based on the length and breadth of the petal. Use cross validation.
3. Compare the tree obtained with the one obtained using the length and width of the sepal. Which tree do you think is the best? Why?
4. Obtain a classification tree to predict the species based on the four variables (length and width of the petal and sepal).
5. Compare the three trees with each other and tell which one is the best for you.

What must the student deliver?

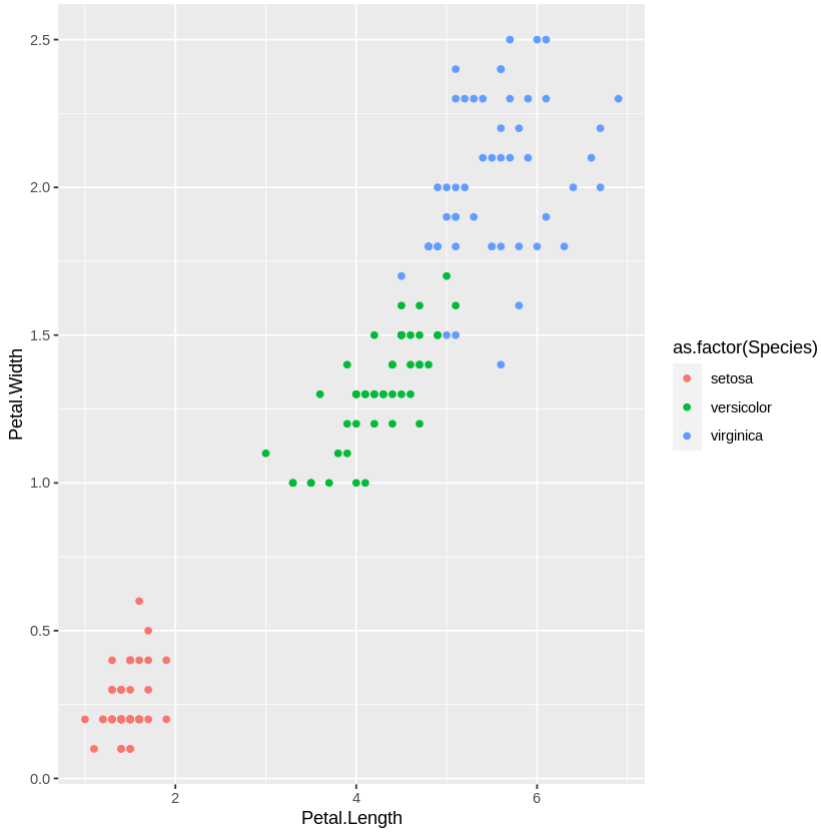
1. Jupyter Notebook (using Colab) that includes the answers to the questions posed and the code used to obtain them.
2. The delivery will be made in the task enabled in the virtual classroom.
3. It is the responsibility of the students to identify and install the necessary packages to manipulate, represent and analyze the data.

Evaluation criteria

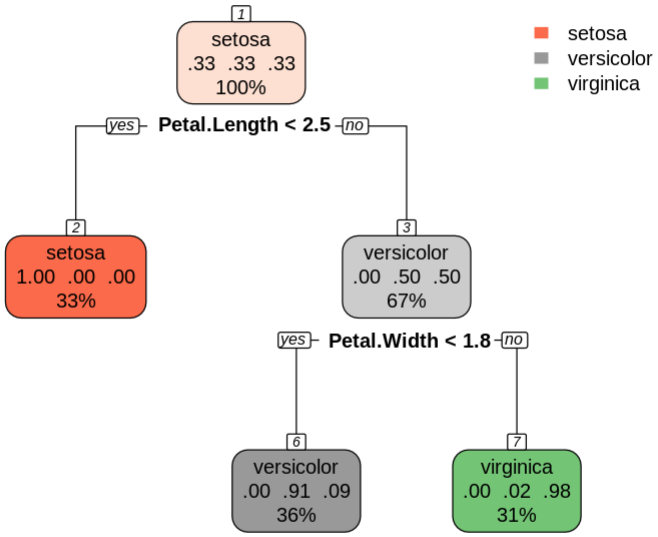
1. Accuracy and coherence between the answers and the statistics obtained after the analysis.
2. Clarity, structure and language used in the report

In [36]:
```
# Task 1

# install.packages ('ggplot2')
#library (ggplot2)
attach ( iris )
ggplot ( data  =  iris ,  mapping  =  aes ( x  =  Petal.Length ,  y  =  Petal.Width ,  co
```

```
The following objects are masked from iris (pos = 3):

    Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species


The following objects are masked from iris (pos = 10):

    Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species


The following objects are masked from iris (pos = 11):

    Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species


The following objects are masked from iris (pos = 12):

    Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species


The following objects are masked from iris (pos = 13):

    Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species


The following objects are masked from iris (pos = 14):

    Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species


The following objects are masked from iris (pos = 15):

    Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, Species
```



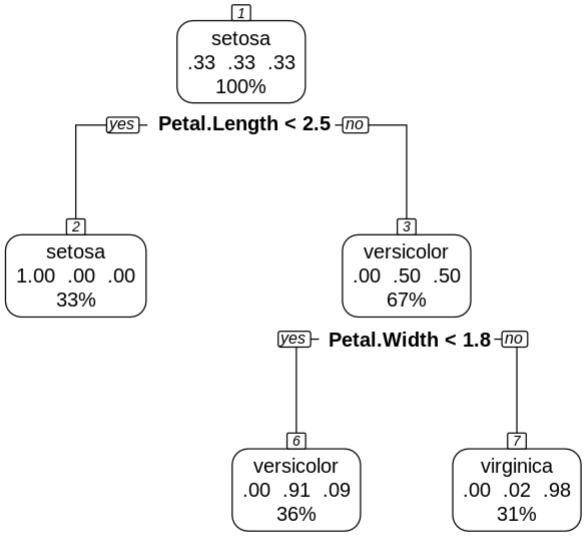
In [20]:
```
#Task 1
iris.tree_task1 <- rpart ( Species ~ Petal.Length + Petal.Width , iris , method
prp ( iris.tree_task1 , type = 2 , extra = "auto" , nn = TRUE , branch = 1 , v
```



In [37]:
```
# Task 2
```

```
train.control <- trainControl ( method = "cv" , number = 10 )
iris.tree_task2 <- train ( Species ~ Petal.Length + Petal.Width , data = iris ,
prp ( iris.tree_task2 $ finalModel , type = 2 , extra = "auto" , nn = TRUE , bra

predicha_task2 <- predict ( iris.tree_task2 $ finalModel , newdata = iris_validacion
confusionMatrix ( predicha_task2 , iris_validacion [[ "Species" ]])
```

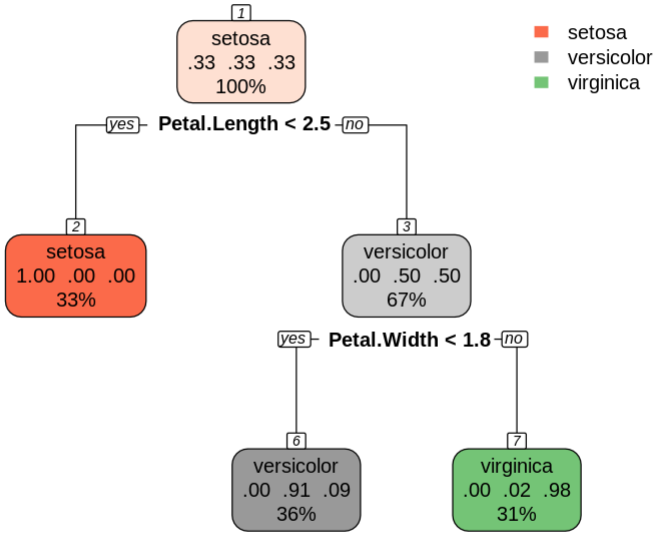
```
Confusion Matrix and Statistics

            Reference
Prediction setosa versicolor virginica
  setosa 12 0 0
  versicolor 0 16 0
  virginica 0 0 16

Overall Statistics

               Accuracy: 1
                 95% CI: (0.9196, 1)
    No Information Rate: 0.3636
    P-Value [Acc> NIR]: <2.2e-16

                  Kappa: 1

 Mcnemar's Test P-Value: NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity 1.0000 1.0000 1.0000
Specificity 1.0000 1.0000 1.0000
Pos Pred Value 1.0000 1.0000 1.0000
Neg Pred Value 1.0000 1.0000 1.0000
Prevalence 0.2727 0.3636 0.3636
Detection Rate 0.2727 0.3636 0.3636
Detection Prevalence 0.2727 0.3636 0.3636
Balanced Accuracy 1.0000 1.0000 1.0000
```



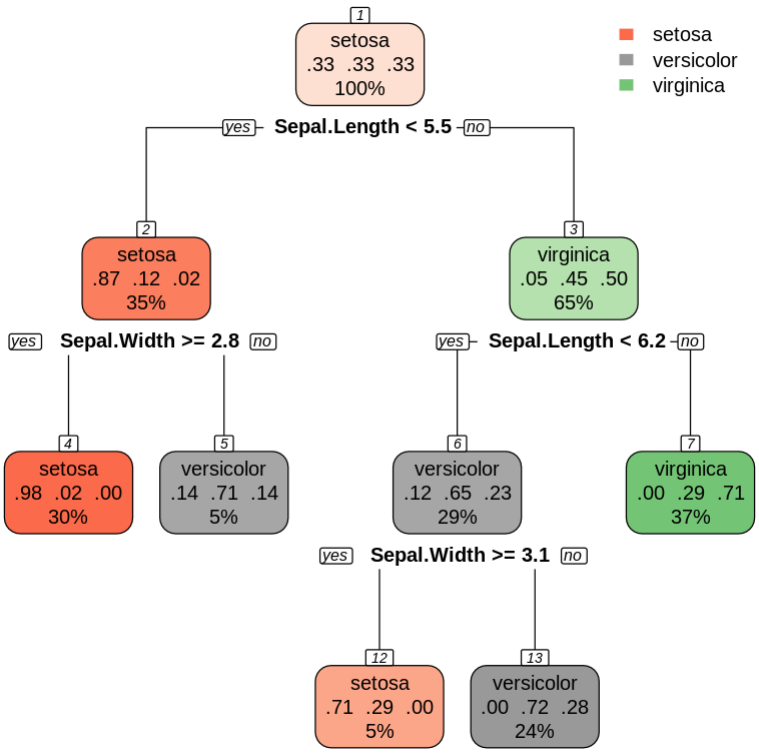
In [80]:
```
# Task 3
```

The accuracy of the model using petal with cross validiation is 0.9556 without two miscalculations. The 95% confidence interval is (0.9314, 0.9946). The accuracy of the sepal is 0.8444 with a 95% confidence interval in (0.7054, 0.9351). This means that the model using the petal information is better with a higher accuracy, less miscalculations and better confidence interval

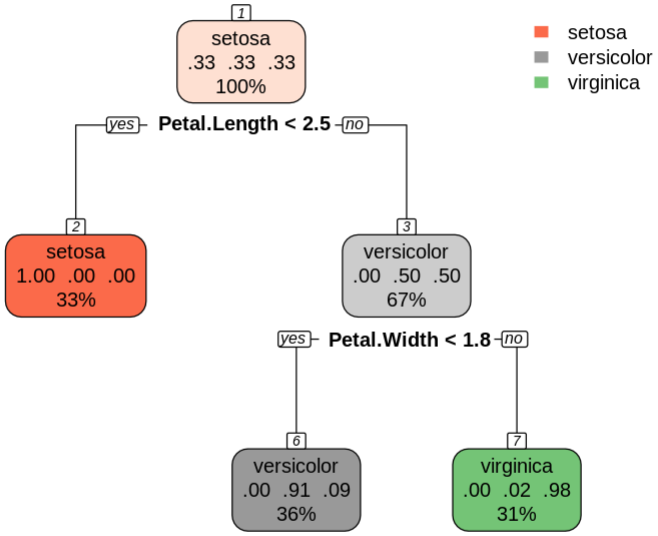
In [38]:
```
# Task 4
iris.tree_task4 <- rpart ( Species ~ Sepal.Length + Sepal.Width + Petal.Length +
prp ( iris.tree_task4 , type = 2 , extra = "auto " , nn = TRUE , branch = 1 ,
```



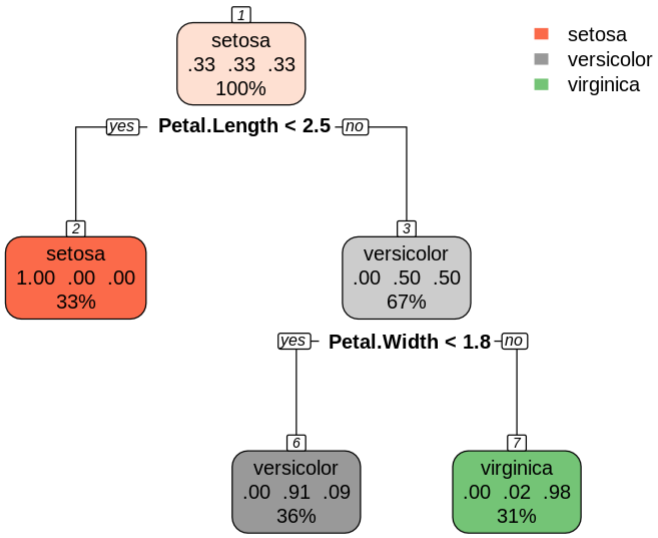
In [39]:
```
# Task 5
prp ( iris.tree1 ,  type  =  2 ,  extra  =  "auto" ,  nn  =  TRUE ,  branch  =  1 ,  varl
```



In [42]:
```
prp ( iris.tree_task1 ,  type  =  2 ,  extra  =  "auto" ,  nn  =  TRUE ,  branch  =  1 ,
```



In [41]:
```
prp ( iris.tree_task4 ,  type  =  2 ,  extra  =  "auto" ,  nn  =  TRUE ,  branch  =  1 ,
```

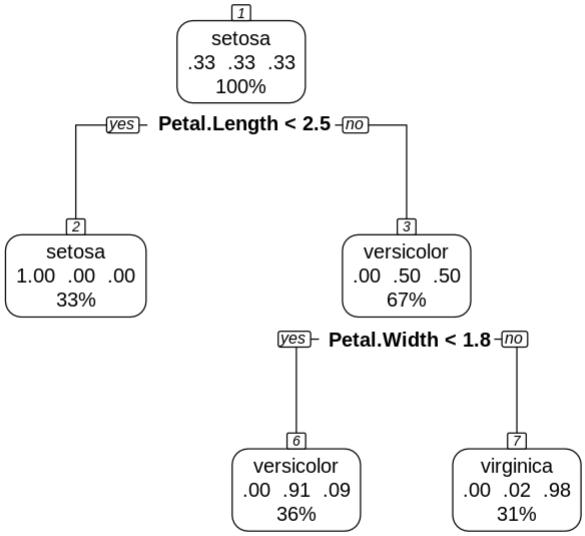


The first tree has more splits so it looks more complicated. The tree including all the variables (tree_task4) is identical to the tree using only petals (tree_task1). This means that the sepal model is more impressive visually but it performs poorly.

In [112]:

```
train.control  <-  trainControl ( method  =  "cv" ,  number  =  10 )
iris.tree_task5  <-  train ( Species  ~  Sepal.Length  +  Sepal.Width  +  Petal.Length  +
prp ( iris.tree_task5 $ finalModel ,  type  =  2 ,  extra =  "auto" ,  nn  =  TRUE ,  bra

predicha_task5  <-  predict ( iris.tree_task5 $ finalModel ,  newdata  =  iris_validacion
confusionMatrix ( predicha_task5 ,  iris_validacion [[ "Species" ]])
```

```
Confusion Matrix and Statistics

            Reference
Prediction setosa versicolor virginica
  setosa 18 0 0
  versicolor 0 16 1
  virginica 0 1 9

Overall Statistics

               Accuracy: 0.9556
                 95% CI: (0.8485, 0.9946)
    No Information Rate: 0.4
    P-Value [Acc> NIR]: 2.842e-15

                  Kappa: 0.9314

 Mcnemar's Test P-Value: NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity 1.0 0.9412 0.9000
Specificity 1.0 0.9643 0.9714
Pos Pred Value 1.0 0.9412 0.9000
Neg Pred Value 1.0 0.9643 0.9714
Prevalence 0.4 0.3778 0.2222
Detection Rate 0.4 0.3556 0.2000
Detection Prevalence 0.4 0.3778 0.2222
Balanced Accuracy 1.0 0.9527 0.9357
```



As we can see from the cross validation results, the including of both petal and sepal does not affect the accuracy from the validation of only including the sepal. The extra information does therefore not benefit the classification model, and it would be smarter to drop the redundant information due to the risk of biased noise to the model. The results are worse for the only sepal model, which is the worse of the three.