```r
# Regression Tree example
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.6.2
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```r
data("msleep")
str(msleep)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    83 obs. of  11 variables:
##  $ name        : chr  "Cheetah" "Owl monkey" "Mountain beaver" "Greater short-tailed shrew" ...
##  $ genus       : chr  "Acinonyx" "Aotus" "Aplodontia" "Blarina" ...
##  $ vore        : chr  "carni" "omni" "herbi" "omni" ...
##  $ order       : chr  "Carnivora" "Primates" "Rodentia" "Soricomorpha" ...
##  $ conservation: chr  "lc" NA "nt" "lc" ...
##  $ sleep_total : num  12.1 17 14.4 14.9 4 14.4 8.7 7 10.1 3 ...
##  $ sleep_rem   : num  NA 1.8 2.4 2.3 0.7 2.2 1.4 NA 2.9 NA ...
##  $ sleep_cycle : num  NA NA NA 0.133 0.667 ...
##  $ awake       : num  11.9 7 9.6 9.1 20 9.6 15.3 17 13.9 21 ...
##  $ brainwt     : num  NA 0.0155 NA 0.00029 0.423 NA NA NA 0.07 0.0982 ...
##  $ bodywt      : num  50 0.48 1.35 0.019 600 ...
```

```r
head(msleep)
```

```
## # A tibble: 6 x 11
##   name  genus vore  order conservation sleep_total sleep_rem sleep_cycle awake
##   <chr> <chr> <chr> <chr> <chr>              <dbl>     <dbl>       <dbl> <dbl>
## 1 Chee~ Acin~ carni Carn~ lc                  12.1      NA        NA      11.9
## 2 Owl ~ Aotus omni  Prim~ <NA>                17         1.8      NA       7
## 3 Moun~ Aplo~ herbi Rode~ nt                  14.4       2.4      NA       9.6
## 4 Grea~ Blar~ omni  Sori~ lc                  14.9       2.3       0.133   9.1
## 5 Cow   Bos   herbi Arti~ domesticated         4         0.7       0.667  20
## 6 Thre~ Brad~ herbi Pilo~ <NA>                14.4       2.2       0.767   9.6
## # ... with 2 more variables: brainwt <dbl>, bodywt <dbl>
```

```r
help("msleep") # read the documentation for the msleep dataset.it is about mammals sleep dataset
```

```
## starting httpd help server ... done
```

```r
# observe the structure of the #msleep dataset
str(data)
```

```
## function (..., list = character(), package = NULL, lib.loc = NULL, verbose = getOption("verbose"),
##     envir = .GlobalEnv, overwrite = TRUE)
```

```r
# creating a new data frame with the following columns included.
mSleepDF1 <- msleep[,c(3,6,10,11)] # 3 = vore ,6=sleep_total, 10=brainwt, 11=bodywt
# observe the structure of the mSleepDF
str(mSleepDF1)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    83 obs. of  4 variables:
##  $ vore       : chr  "carni" "omni" "herbi" "omni" ...
##  $ sleep_total: num  12.1 17 14.4 14.9 4 14.4 8.7 7 10.1 3 ...
##  $ brainwt    : num  NA 0.0155 NA 0.00029 0.423 NA NA NA 0.07 0.0982 ...
##  $ bodywt     : num  50 0.48 1.35 0.019 600 ...
```

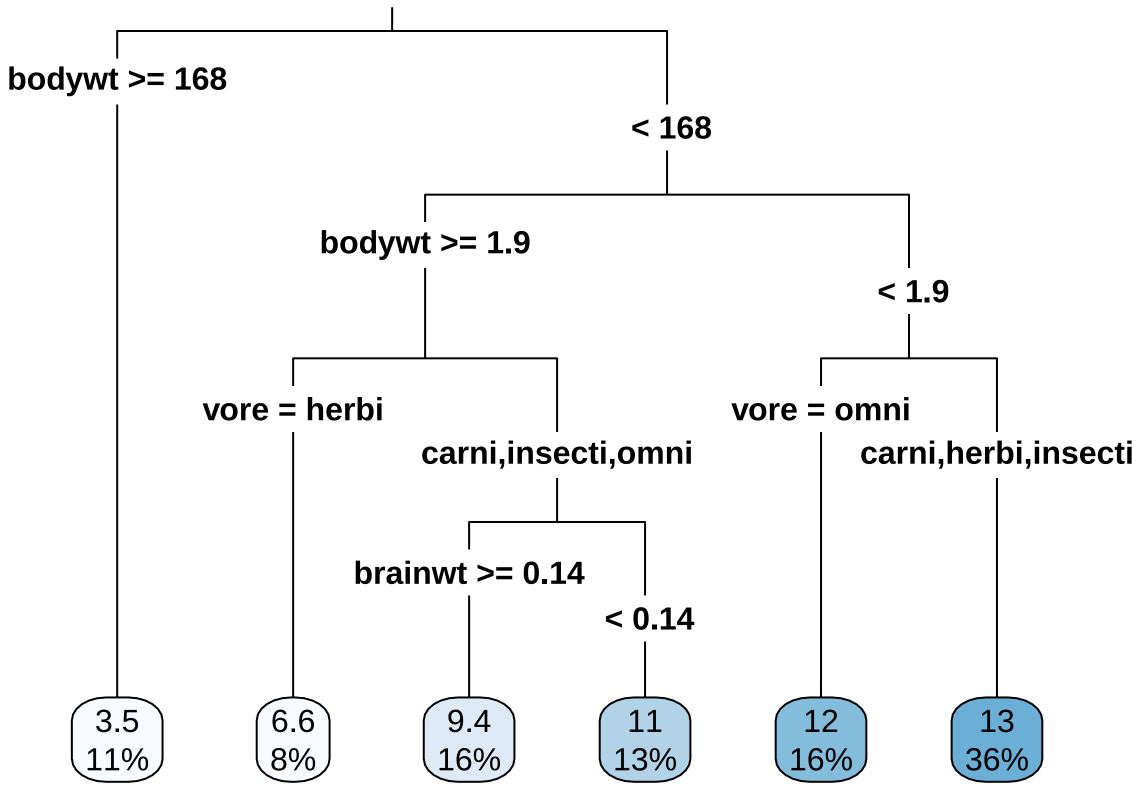
```
head(mSleepDF1)
```

```
## # A tibble: 6 x 4
##   vore  sleep_total  brainwt  bodywt
##   <chr>       <dbl>    <dbl>   <dbl>
## 1 carni        12.1 NA        50
## 2 omni         17    0.0155    0.48
## 3 herbi        14.4 NA         1.35
## 4 omni         14.9  0.00029   0.019
## 5 herbi         4    0.423   600
## 6 herbi        14.4 NA         3.85
```

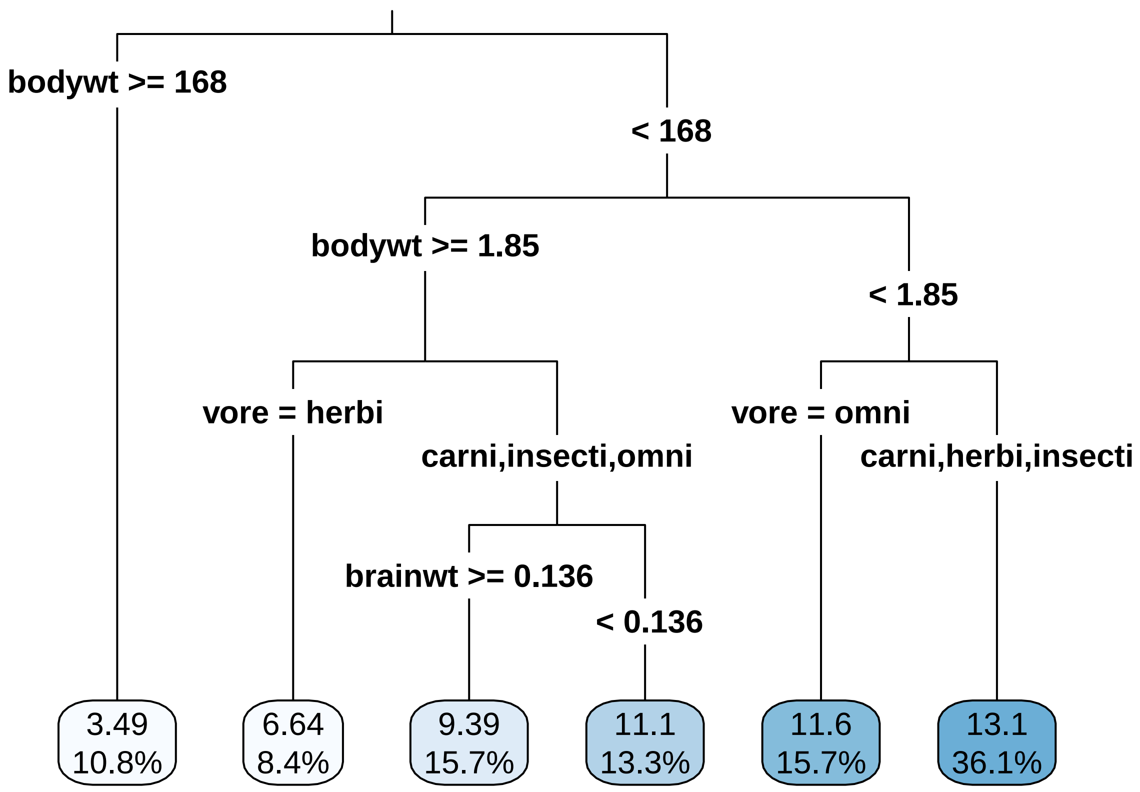
```
# Building Regression Decision Tree that #predicts the total sleeping
# hours of the mamals based on the other #variables available on the dataset
help("rpart") # Read the documentation for the rpart() function.
sleepModel_1 <- rpart(sleep_total ~ ., data=mSleepDF1, method = "anova")
# method we are using here is anova becuase our target here is sleep_total is a numerical one.
sleepModel_1
```

```
## n= 83
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 83 1624.066000 10.433730
##    2) bodywt>=167.947 9    7.868889  3.488889 *
##    3) bodywt< 167.947 74 1129.325000 11.278380
##      6) bodywt>=1.85 31  458.593500  9.361290
##       12) vore=herbi 7   88.337140  6.642857 *
##       13) vore=carni,insecti,omni 24  303.439600 10.154170
##         26) brainwt>=0.136 13  128.669200  9.392308 *
##         27) brainwt< 0.136 11  158.307300 11.054550 *
##      7) bodywt< 1.85 43  474.662800 12.660470
##       14) vore=omni 13  141.370800 11.638460 *
##       15) vore=carni,herbi,insecti 30  313.829700 13.103330 *
```

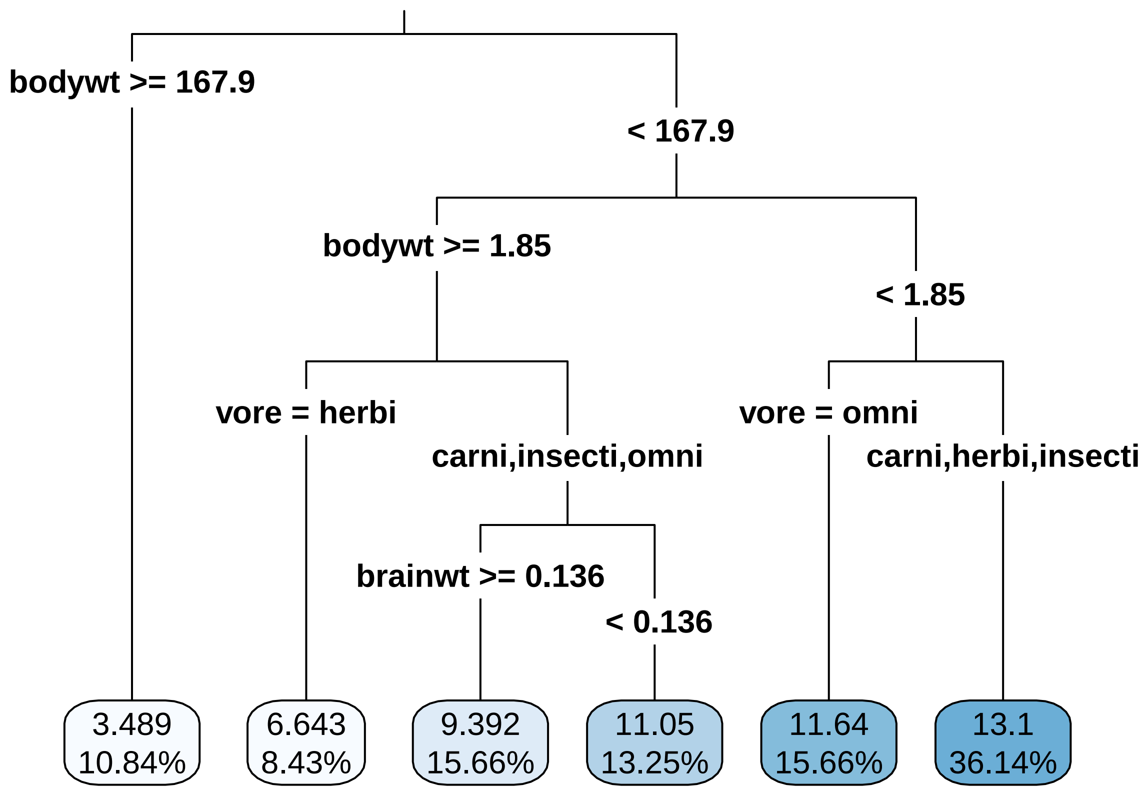
```
# let's visualize this using rpart.plot()
help("rpart.plot")
rpart.plot(sleepModel_1, type = 3, fallen.leaves = TRUE)
```



```
# type = 3, Draw separate split labels for the left and right directions.See the documentation
#fallen.leaves = TRUE,  Default TRUE to position the leaf nodes at the bottom of the graph.
#It can be helpful to use FALSE if the graph is too crowded and the text size is too small.
rpart.plot(sleepModel_1, type = 3,digits = 3, fallen.leaves = TRUE) # with 3 digits
```



```
rpart.plot(sleepModel_1, type = 3,digits = 4, fallen.leaves = TRUE)
```



```r
#Classification Tree example
# instrall the C50 package
#install.packages("C50")
require(C50)
```

```
## Loading required package: C50
```

```
## Warning: package 'C50' was built under R version 3.6.2
```

```r
# we will be using the iris dataset to do a #classfication
data("iris")
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```r
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
table(iris$Species)
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

```r
# set the seed
set.seed(9850)
# generate random numbers
help("runif")
grn <-runif(nrow(iris))

# creating a randomized iris dataset ,  shuffling the dataset
# we use the order() function along with the #random numbers we generated.
irisrand <-iris[order(grn),]

# obsrve that rows are now randomly shuffled.
str(irisrand)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  7.1 5.1 6 5.4 5.8 6.9 7.7 5.5 5.7 4.4 ...
##  $ Sepal.Width : num  3 3.8 2.2 3.9 2.7 3.1 3.8 2.6 2.6 3.2 ...
##  $ Petal.Length: num  5.9 1.5 4 1.3 3.9 4.9 6.7 4.4 3.5 1.3 ...
##  $ Petal.Width : num  2.1 0.3 1 0.4 1.2 1.5 2.2 1.2 1 0.2 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 3 1 2 1 2 2 3 2 2 1 ...
```

```r
help("c5.0")
```

```
## No documentation for 'c5.0' in specified packages and libraries:
## you could try '??c5.0'
```

```r
classificationmodel1 <-C5.0(irisrand[1:100,-5], irisrand[1:100,5])
classificationmodel1
```

```
##
## Call:
## C5.0.default(x = irisrand[1:100, -5], y = irisrand[1:100, 5])
##
## Classification Tree
## Number of samples: 100
## Number of predictors: 4
##
## Tree size: 4
##
## Non-standard options: attempt to group attributes
```

```r
summary(classificationmodel1)
```

```
##
## Call:
## C5.0.default(x = irisrand[1:100, -5], y = irisrand[1:100, 5])
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Feb 24 14:20:10 2020
## -------------------------------
##
## Class specified by attribute `outcome'
```

```
##
## Read 100 cases (5 attributes) from undefined.data
##
## Decision tree:
##
## Petal.Length <= 1.9: setosa (34)
## Petal.Length > 1.9:
## :...Petal.Width > 1.6: virginica (29)
##     Petal.Width <= 1.6:
##     :...Petal.Length <= 4.9: versicolor (35)
##         Petal.Length > 4.9: virginica (2)
##
##
## Evaluation on training data (100 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       4    0( 0.0%)   <<
##
##
##     (a)   (b)   (c)    <-classified as
##    ----  ----  ----
##      34                (a): class setosa
##            35          (b): class versicolor
##                  31    (c): class virginica
##
##
##  Attribute usage:
##
##  100.00% Petal.Length
##   66.00% Petal.Width
##
##
## Time: 0.0 secs
```

```
# now we will do the prediction using the #predict() function
# We are using the remaining last 50 rows for #here starting from 101 row to 150th row
prediction1 <- predict(classificationmodel1,irisrand[101:150,])
prediction1
```

```
##  [1] virginica  setosa     versicolor virginica  versicolor setosa
##  [7] setosa     versicolor versicolor versicolor versicolor virginica
## [13] virginica  setosa     versicolor virginica  virginica  virginica
## [19] versicolor virginica  setosa     virginica  virginica  setosa
## [25] virginica  setosa     setosa     versicolor setosa     versicolor
## [31] setosa     virginica  virginica  virginica  setosa     virginica
## [37] versicolor virginica  setosa     setosa     virginica  setosa
## [43] virginica  virginica  virginica  setosa     virginica  virginica
## [49] versicolor setosa
## Levels: setosa versicolor virginica
```

```
# we will use the confusion matrix to #understand our prediction
# Read the documentation for the table() function in RStudio help
```

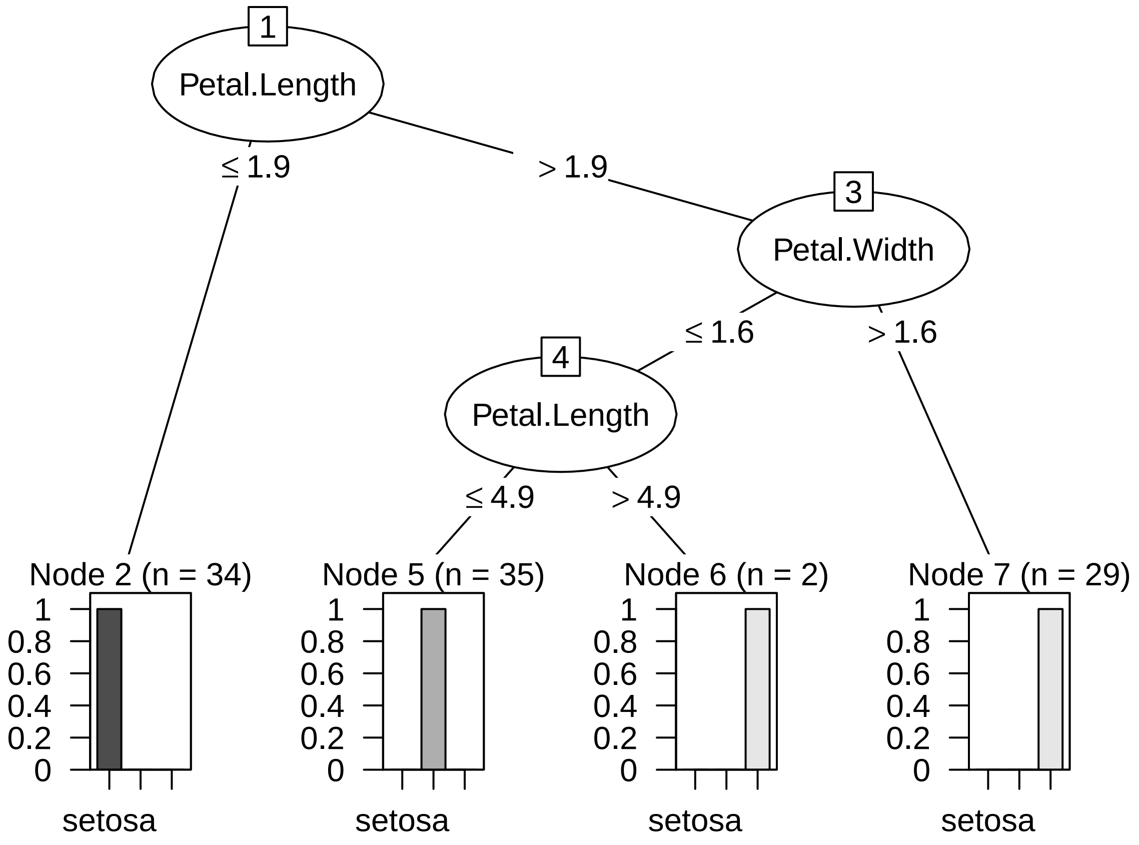
```r
table(irisrand[101:150,5],prediction1)
```

```
##             prediction1
##              setosa versicolor virginica
##   setosa         16          0         0
##   versicolor      0         12         3
##   virginica       0          0        19
```

```r
# you can write the same above line by defining what is the "predicted"
##  table(irisrand[101:150,5],Predicted = prediction1)

# we will use the confusion matrix to  #understand our prediction
# Read the documentation for the table() #function in RStudio help
table(irisrand[101:150,5],prediction1)
```

```
##             prediction1
##              setosa versicolor virginica
##   setosa         16          0         0
##   versicolor      0         12         3
##   virginica       0          0        19
```

```r
# We can plot the classification model tree #using the plot() function
plot(classificationmodel1)
```



```r
## Call the NaiveBayes Classifier Package e1071, which auto calls the Class package ##
library("e1071")
```

```
## Warning: package 'e1071' was built under R version 3.6.2
```

```r
classifier<-naiveBayes(iris[,1:4], iris[,5])
table(predict(classifier, iris[,-5]), iris[,5], dnn=list('predicted','actual'))
```

```
##               actual
## predicted    setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         47         3
##   virginica       0          3        47
```

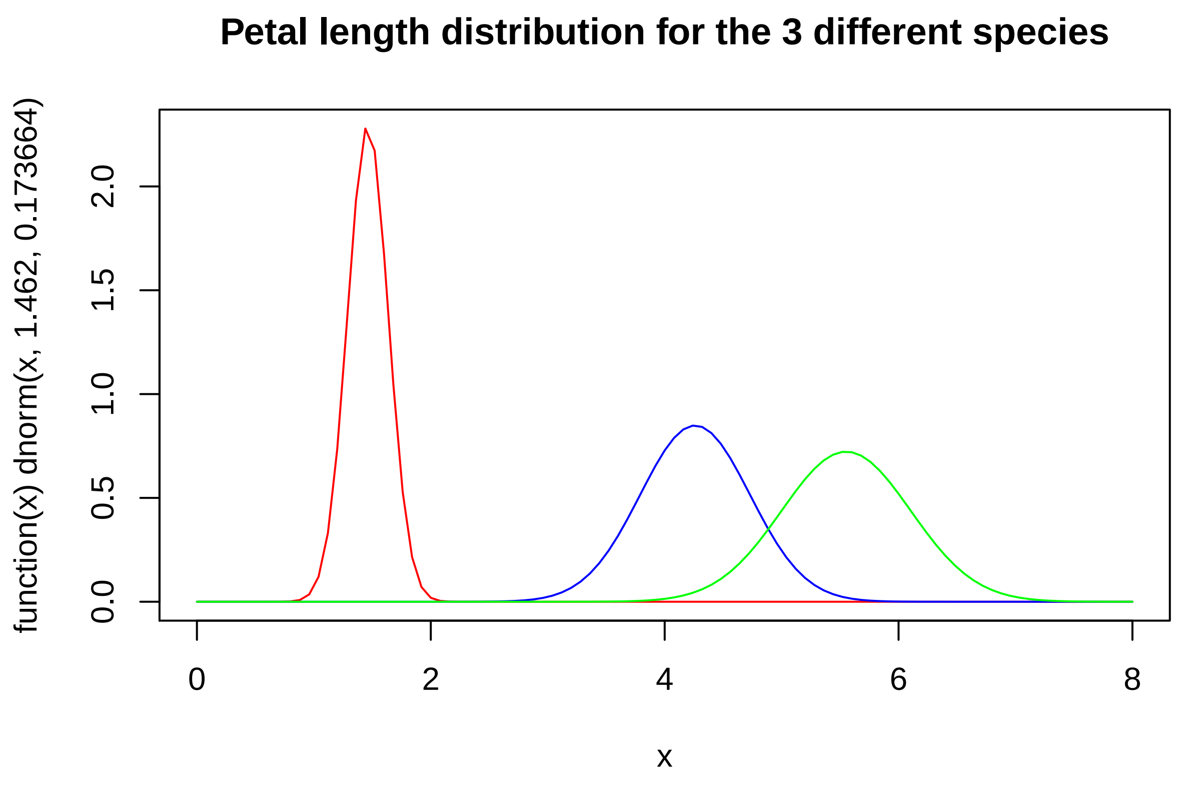
```r
classifier$apriori
```

```
## iris[, 5]
##     setosa versicolor  virginica
##         50         50         50
```

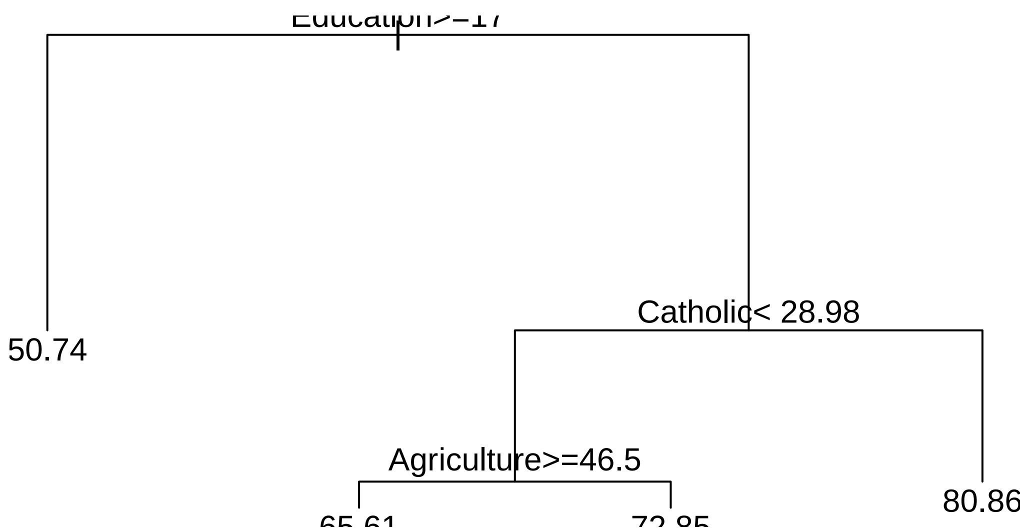
```r
classifier$tables$Petal.Length
```

```
##               Petal.Length
## iris[, 5]      [,1]      [,2]
##   setosa     1.462 0.1736640
##   versicolor 4.260 0.4699110
##   virginica  5.552 0.5518947
```

```r
plot(function(x) dnorm(x, 1.462, 0.1736640), 0, 8, col="red", main="Petal length distribution for the 3
curve(dnorm(x, 4.260, 0.4699110), add=TRUE, col="blue")
curve(dnorm(x, 5.552, 0.5518947 ), add=TRUE, col = "green")
```



```r
#ctree1
require(rpart)
Swiss_rpart <- rpart(Fertility ~ Agriculture + Education + Catholic, data = swiss)
plot(Swiss_rpart) # try some different plot options
text(Swiss_rpart) # try some different text options
```

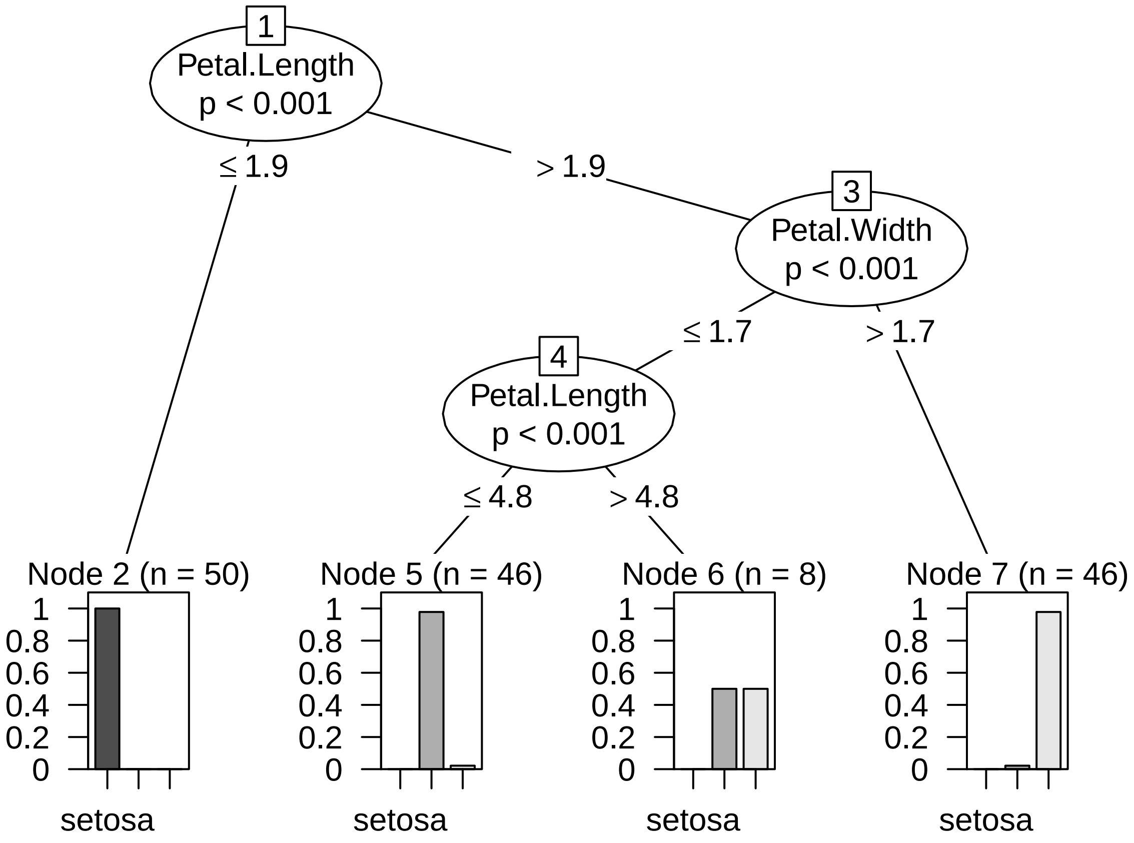


```r
require(party)
```

```
## Loading required package: party

## Warning: package 'party' was built under R version 3.6.2

## Loading required package: grid
## Loading required package: mvtnorm
## Loading required package: modeltools
## Loading required package: stats4
## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 3.6.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.6.2

##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
```

```
##     as.Date, as.Date.numeric
##
## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 3.6.2
```

```r
treeSwiss<-ctree(Species ~ ., data=iris)
plot(treeSwiss)
```



```r
cforest(Species ~ ., data=iris, controls=cforest_control(mtry=2, mincriterion=0))
```

```
##
##   Random Forest using Conditional Inference Trees
##
## Number of trees:  500
##
## Response:  Species
## Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
## Number of observations:  150
```

```r
treeFert<-ctree(Fertility ~ Agriculture + Education + Catholic, data = swiss)
cforest(Fertility ~ Agriculture + Education + Catholic, data = swiss, controls=cforest_control(mtry=2, 
```

```
##
##   Random Forest using Conditional Inference Trees
##
## Number of trees:  500
##
```

```
## Response:  Fertility
## Inputs:  Agriculture, Education, Catholic
## Number of observations:  47
```

```
# look at help info, vary parameters.

#install.packages("tree")
library(tree)
```

```
## Warning: package 'tree' was built under R version 3.6.2
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

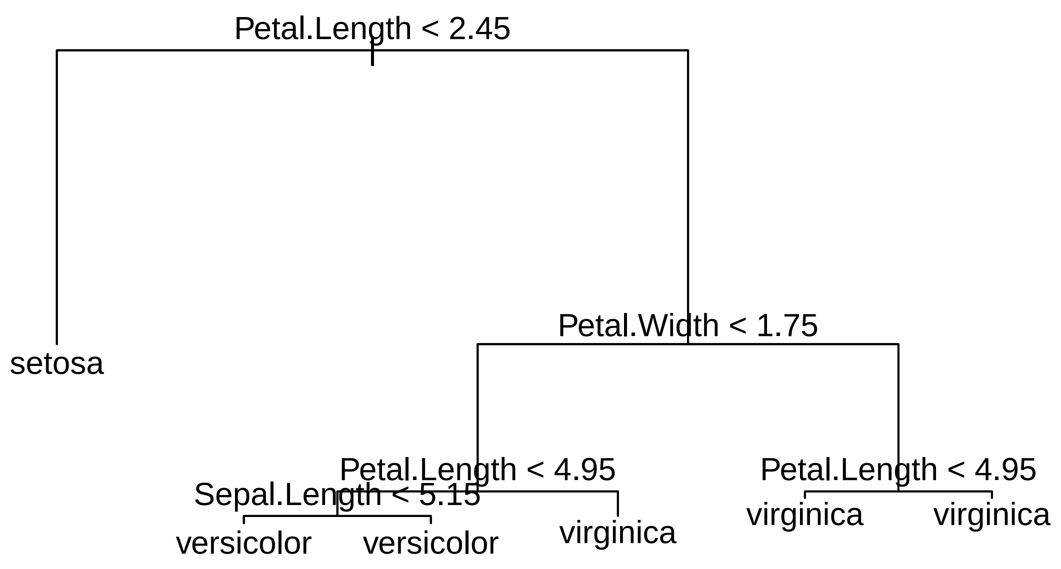
```
tr <- tree(Species ~ ., data=iris)
tr
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 150 329.600 setosa ( 0.33333 0.33333 0.33333 )
##    2) Petal.Length < 2.45 50   0.000 setosa ( 1.00000 0.00000 0.00000 ) *
##    3) Petal.Length > 2.45 100 138.600 versicolor ( 0.00000 0.50000 0.50000 )
##      6) Petal.Width < 1.75 54  33.320 versicolor ( 0.00000 0.90741 0.09259 )
##       12) Petal.Length < 4.95 48   9.721 versicolor ( 0.00000 0.97917 0.02083 )
##         24) Sepal.Length < 5.15 5   5.004 versicolor ( 0.00000 0.80000 0.20000 ) *
##         25) Sepal.Length > 5.15 43   0.000 versicolor ( 0.00000 1.00000 0.00000 ) *
##       13) Petal.Length > 4.95 6   7.638 virginica ( 0.00000 0.33333 0.66667 ) *
##      7) Petal.Width > 1.75 46   9.635 virginica ( 0.00000 0.02174 0.97826 )
##       14) Petal.Length < 4.95 6   5.407 virginica ( 0.00000 0.16667 0.83333 ) *
##       15) Petal.Length > 4.95 40   0.000 virginica ( 0.00000 0.00000 1.00000 ) *
```

```
tr$frame
```

```
##             var   n        dev       yval splits.cutleft splits.cutright
## 1  Petal.Length 150 329.583687     setosa          <2.45           >2.45
## 2        <leaf>  50   0.000000     setosa
## 3   Petal.Width 100 138.629436 versicolor          <1.75           >1.75
## 6  Petal.Length  54  33.317509 versicolor          <4.95           >4.95
## 12 Sepal.Length  48   9.721422 versicolor          <5.15           >5.15
## 24       <leaf>   5   5.004024 versicolor
## 25       <leaf>  43   0.000000 versicolor
## 13       <leaf>   6   7.638170  virginica
## 7  Petal.Length  46   9.635384  virginica          <4.95           >4.95
## 14       <leaf>   6   5.406735  virginica
## 15       <leaf>  40   0.000000  virginica
##    yprob.setosa yprob.versicolor yprob.virginica
## 1    0.33333333       0.33333333      0.33333333
## 2    1.00000000       0.00000000      0.00000000
## 3    0.00000000       0.50000000      0.50000000
## 6    0.00000000       0.90740741      0.09259259
## 12   0.00000000       0.97916667      0.02083333
## 24   0.00000000       0.80000000      0.20000000
## 25   0.00000000       1.00000000      0.00000000
## 13   0.00000000       0.33333333      0.66666667
## 7    0.00000000       0.02173913      0.97826087
## 14   0.00000000        0.16666667       0.83333333
## 15   0.00000000        0.00000000       1.00000000
```

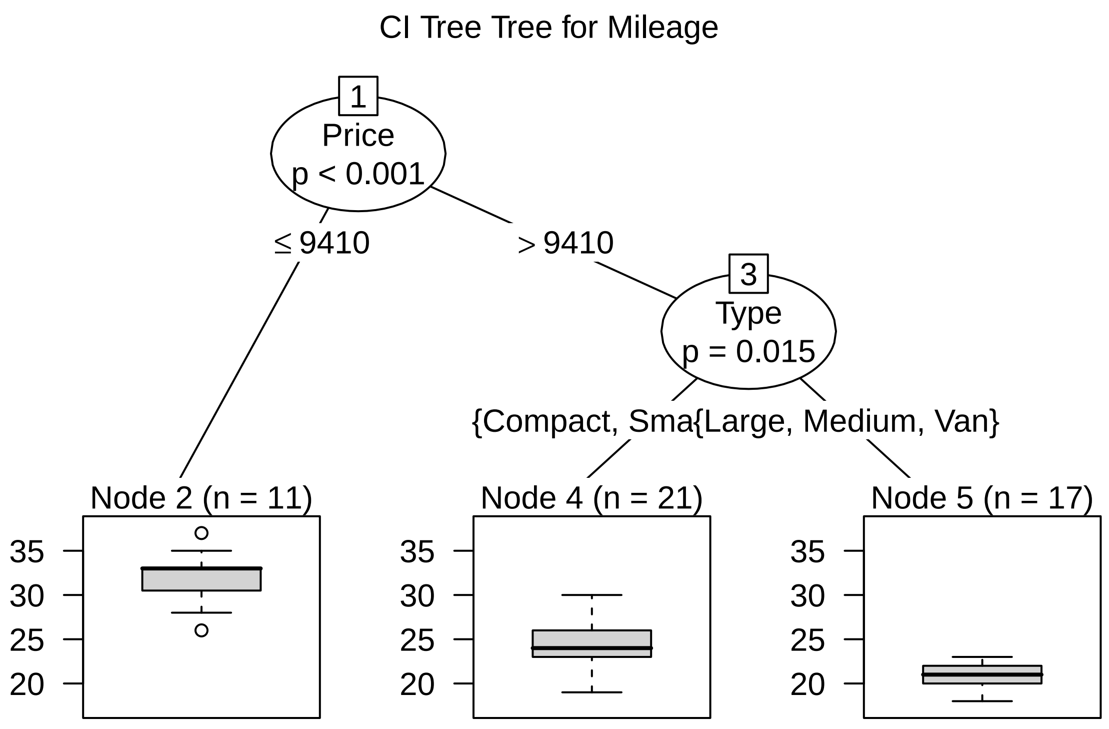
```
plot(tr)
text(tr)
```



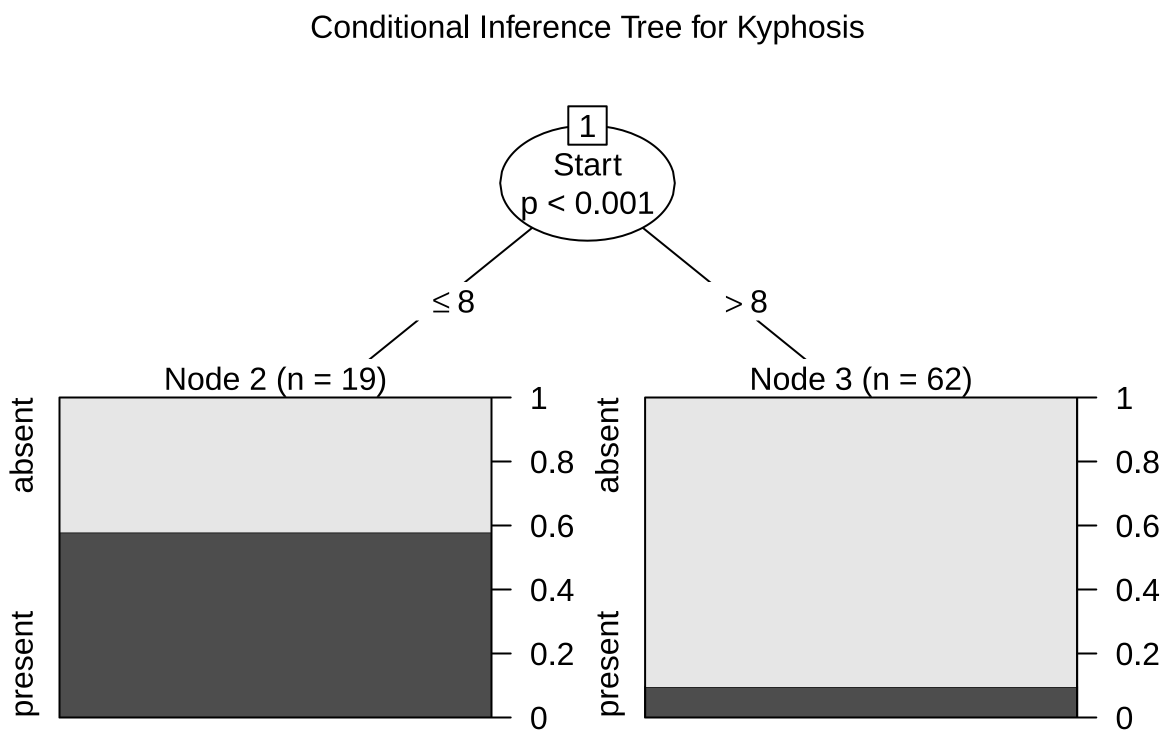
```
#find "prettier" ways to plot the tree

#ctree2
# Conditional Inference Tree for Mileage
fit2M <- ctree(Mileage~Price + Country + Reliability + Type, data=na.omit(cu.summary))
summary(fit2M)
```

```
##     Length      Class       Mode
##          1 BinaryTree         S4
```

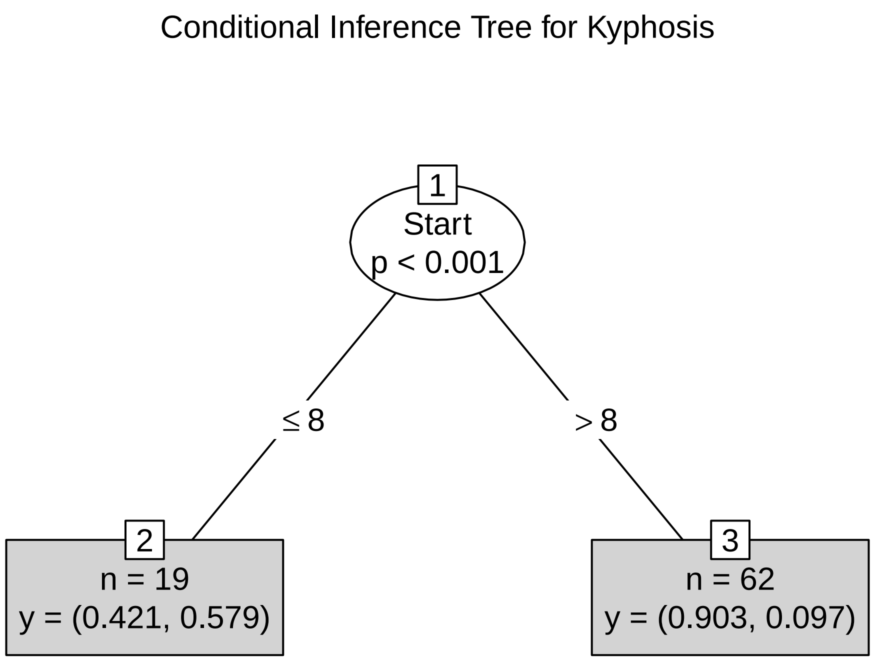
```
# plot tree
plot(fit2M, uniform=TRUE, main="CI Tree Tree for Mileage ")
```



```
#text(fit2M, use.n=TRUE, all=TRUE, cex=.8)

#ctree3
fitK <- ctree(Kyphosis ~ Age + Number + Start, data=kyphosis)
plot(fitK, main="Conditional Inference Tree for Kyphosis")
```



```
plot(fitK, main="Conditional Inference Tree for Kyphosis",type="simple")
```


Conditional Inference Tree for Kyphosis
1
Start
p < 0.001
≤ 8
> 8
2
n = 19
y = (0.421, 0.579)
3
n = 62
y = (0.903, 0.097)

```
#etc.
```