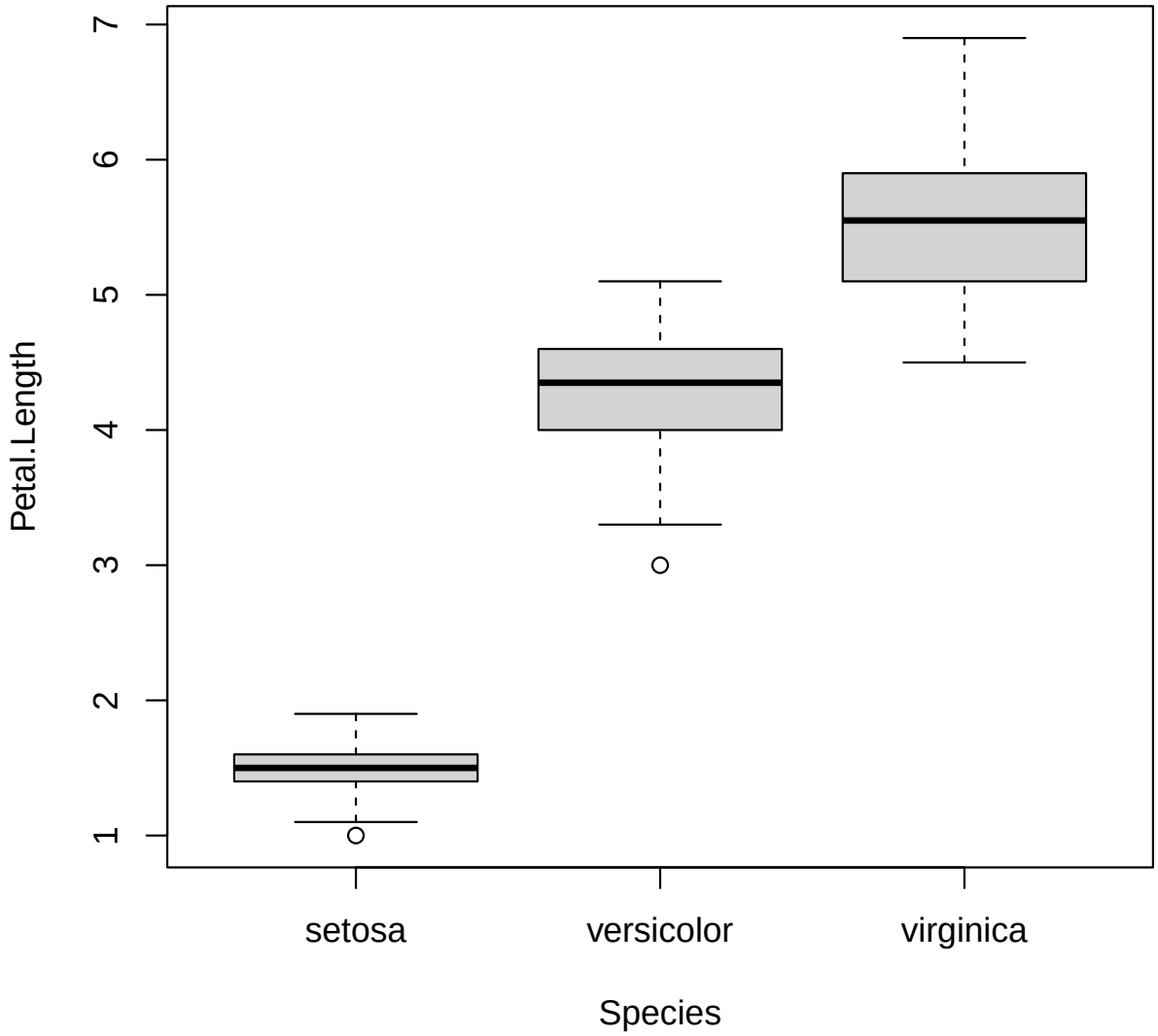
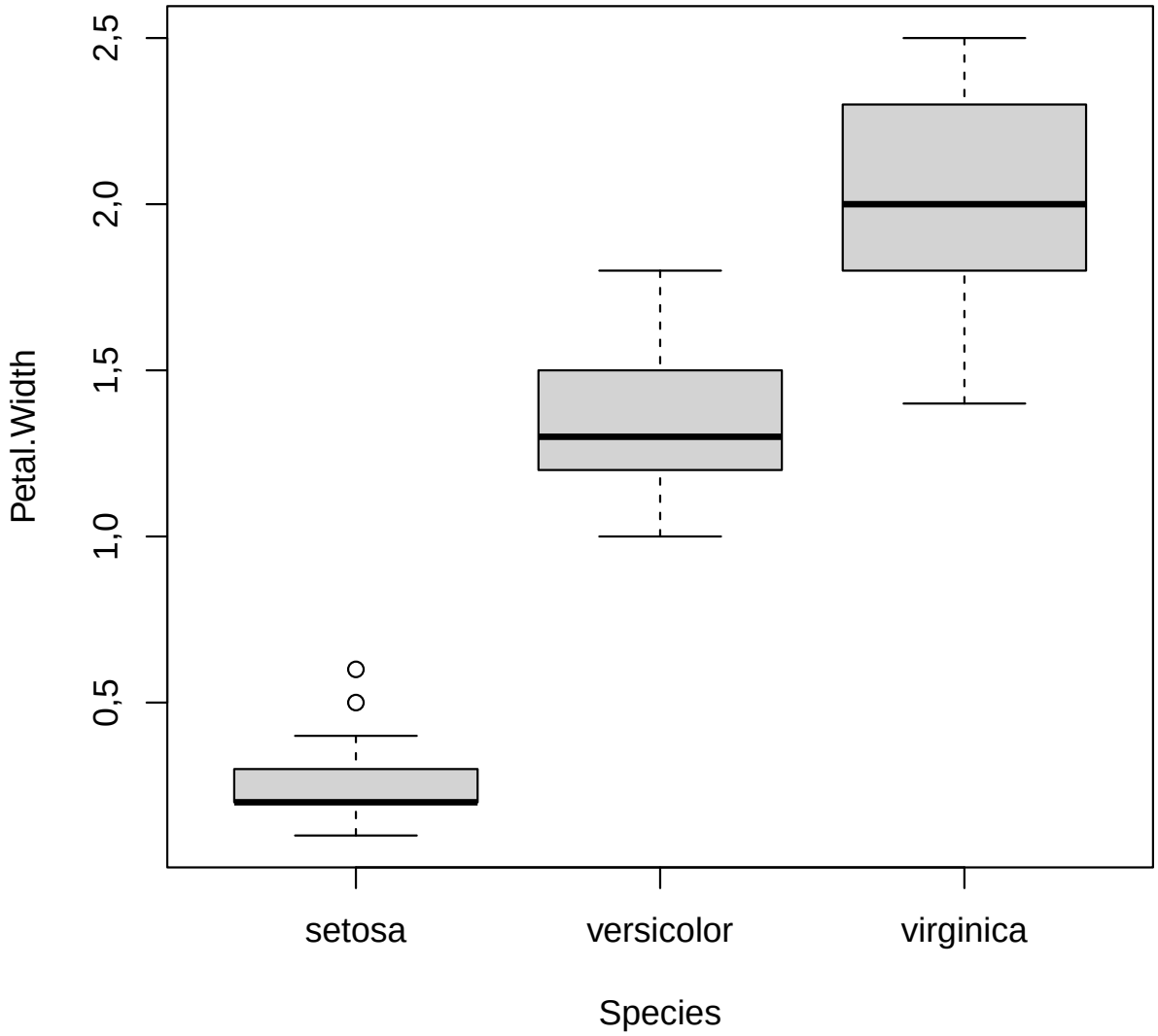
Árboles de decisión

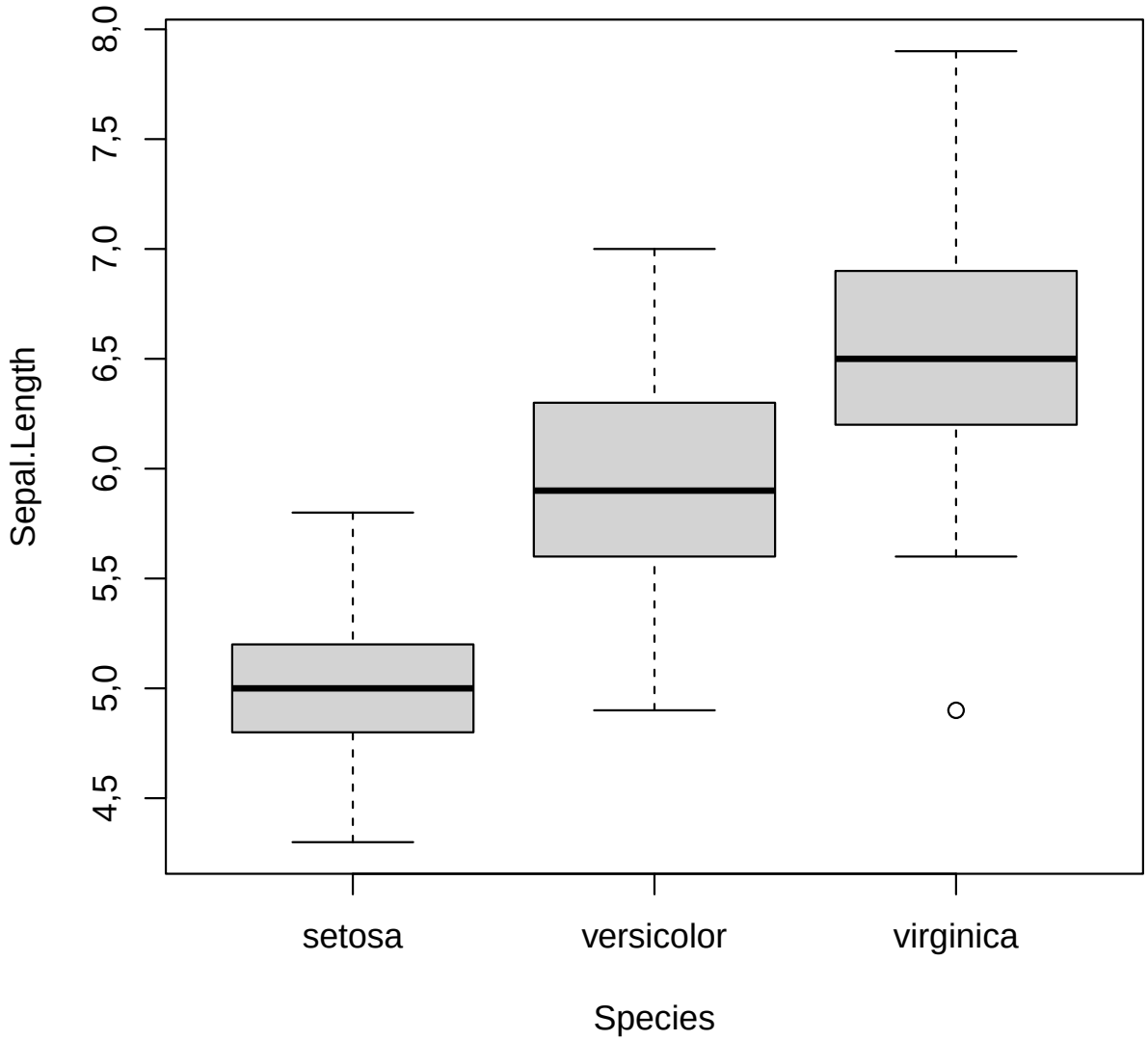
N. Corral,
C. Carleos

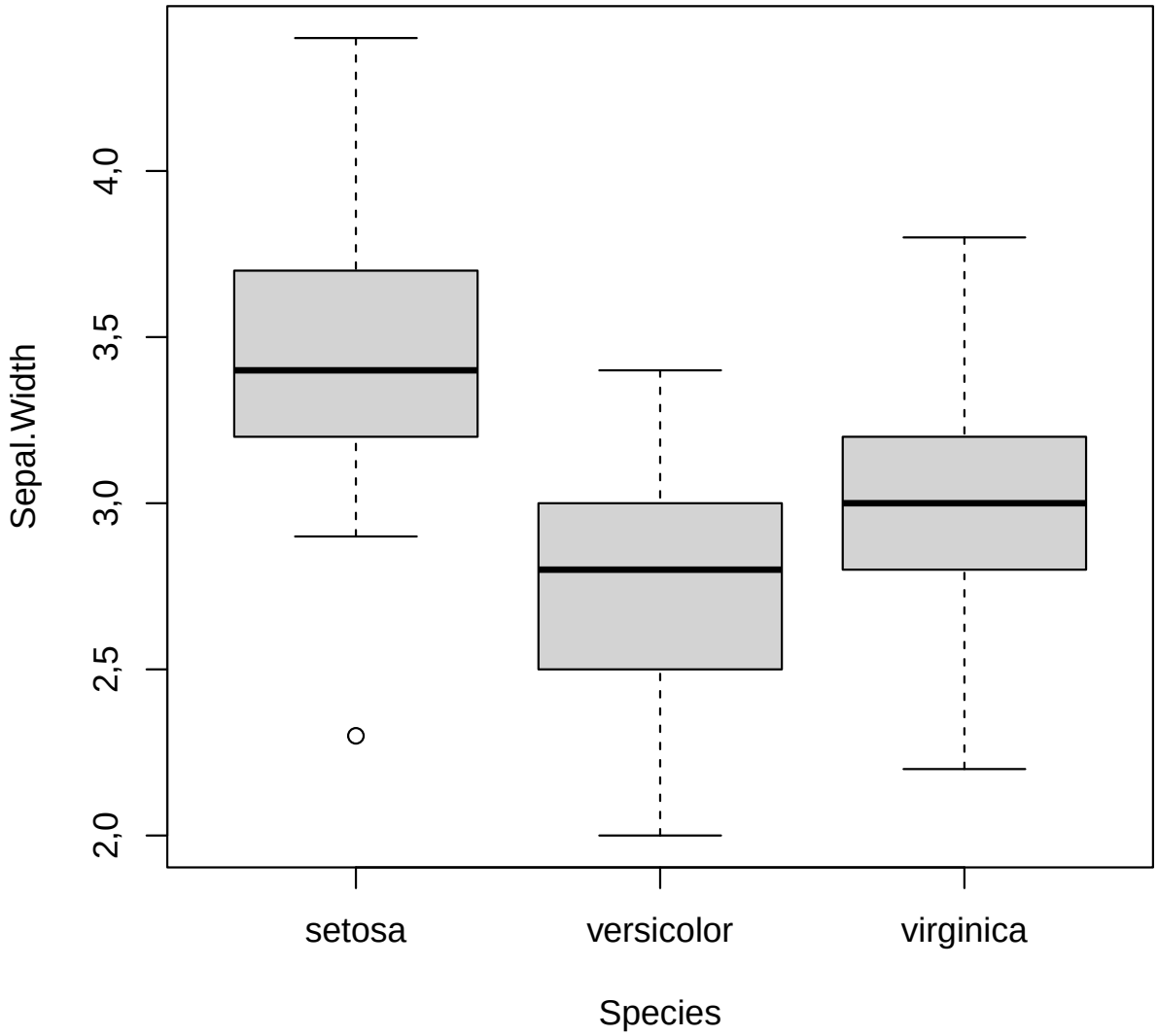
19 de abril de 2022

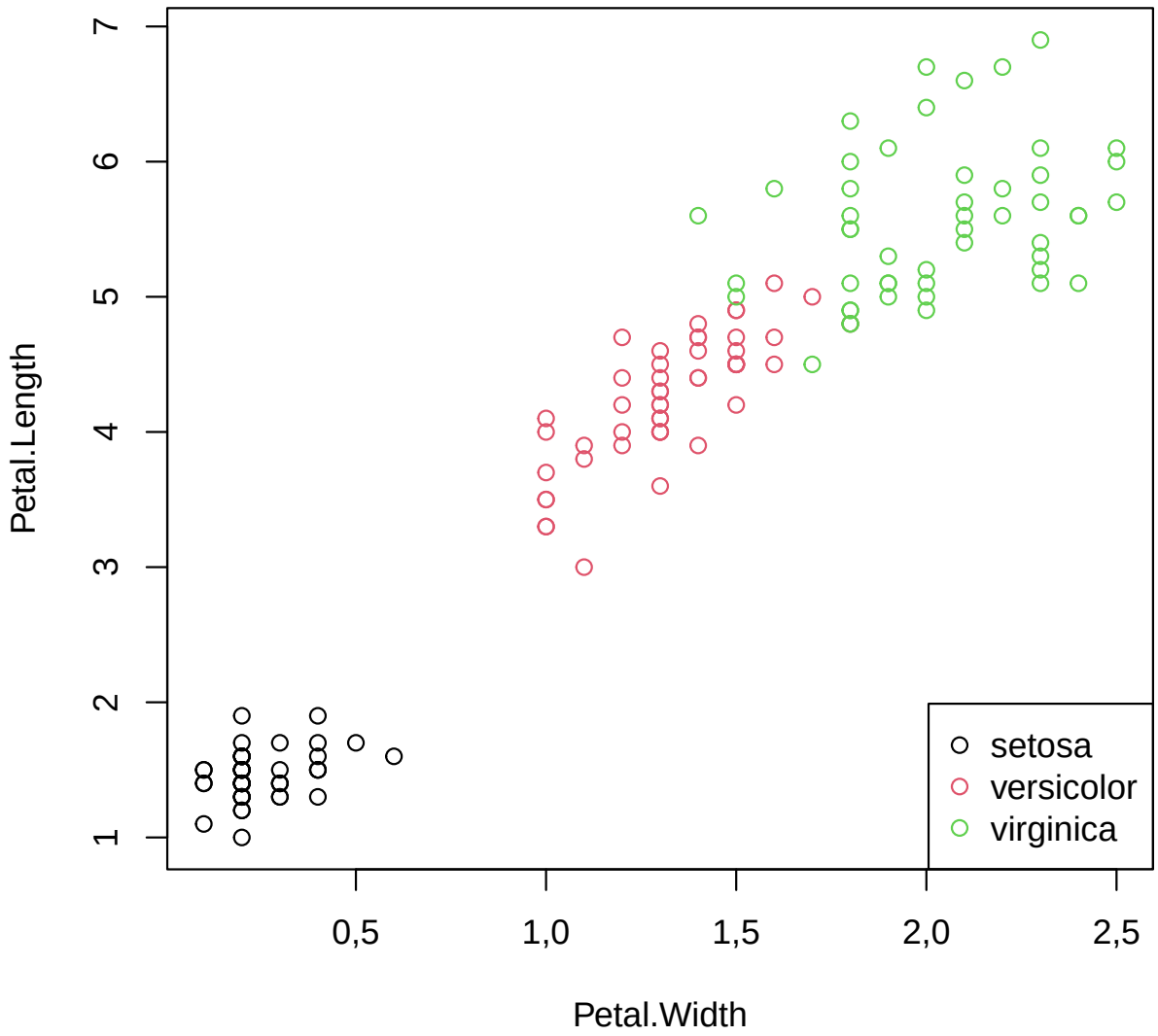
```
> summary (iris)
  Sepal.Length    Sepal.Width     Petal.Length
 Min.   :4,30   Min.   :2,00   Min.   :1,00
 1st Qu.:5,10   1st Qu.:2,80   1st Qu.:1,60
 Median :5,80   Median :3,00   Median :4,35
 Mean   :5,84   Mean   :3,06   Mean   :3,76
 3rd Qu.:6,40   3rd Qu.:3,30   3rd Qu.:5,10
 Max.   :7,90   Max.   :4,40   Max.   :6,90
  Petal.Width          Species
 Min.   :0,1   setosa    :50
 1st Qu.:0,3   versicolor:50
 Median :1,3   virginica :50
 Mean   :1,2
 3rd Qu.:1,8
 Max.   :2,5
```

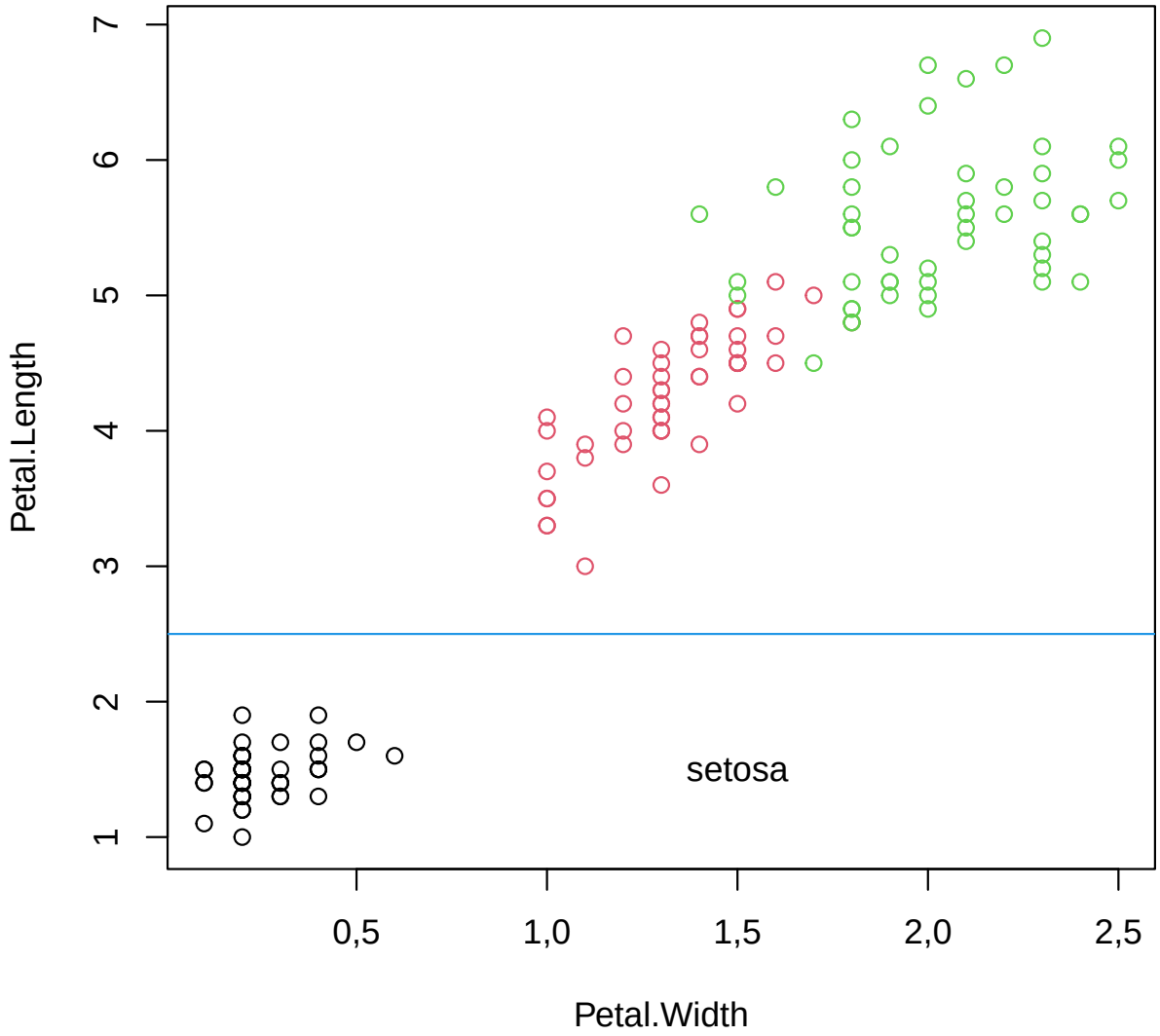

Petal.Length
1
2
3
4
5
6
7
setosa
versicolor
virginica
Species

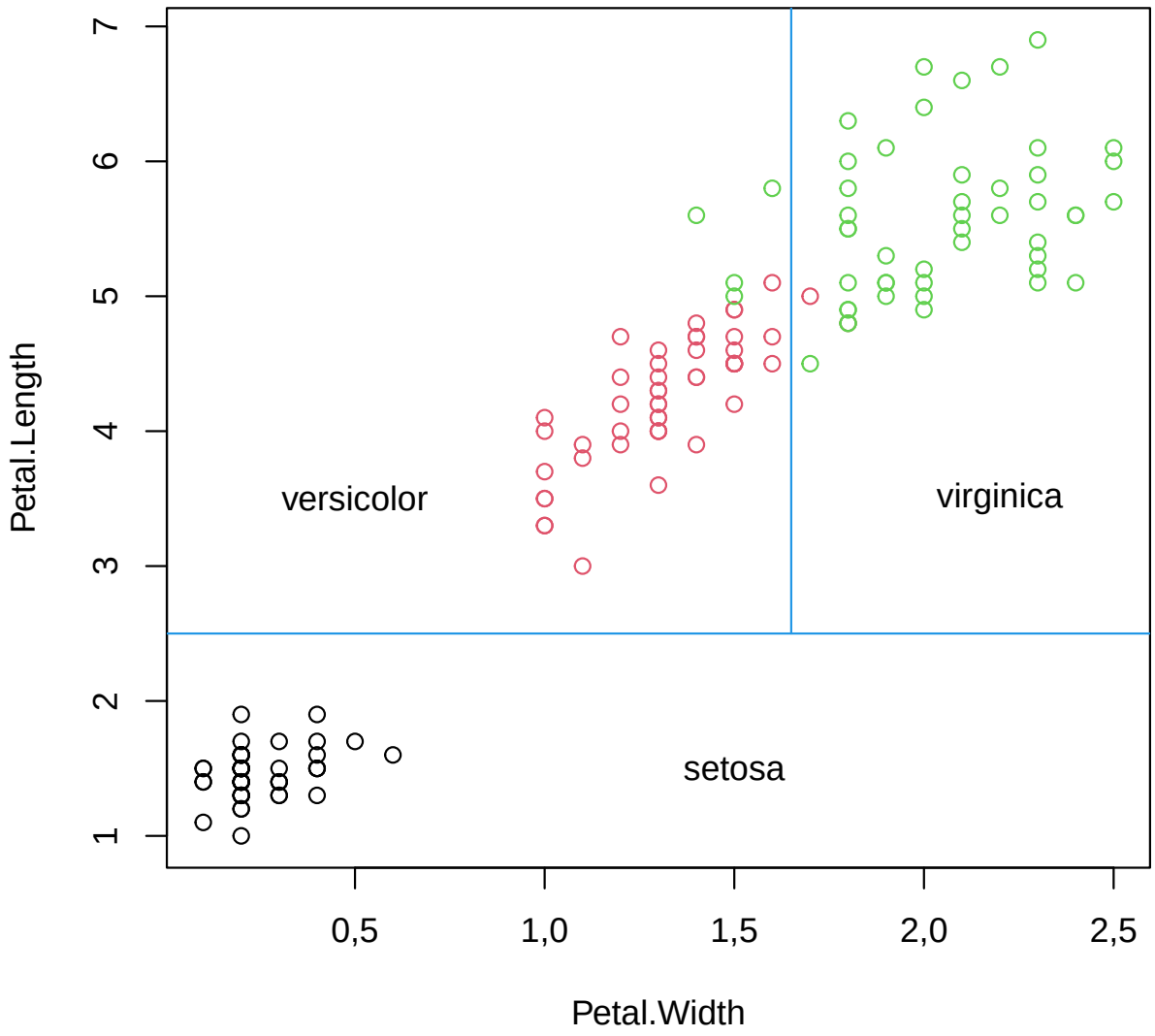

Petal.Width
0,5
1,0
1,5
2,0
2,5
setosa
versicolor
virginica
Species

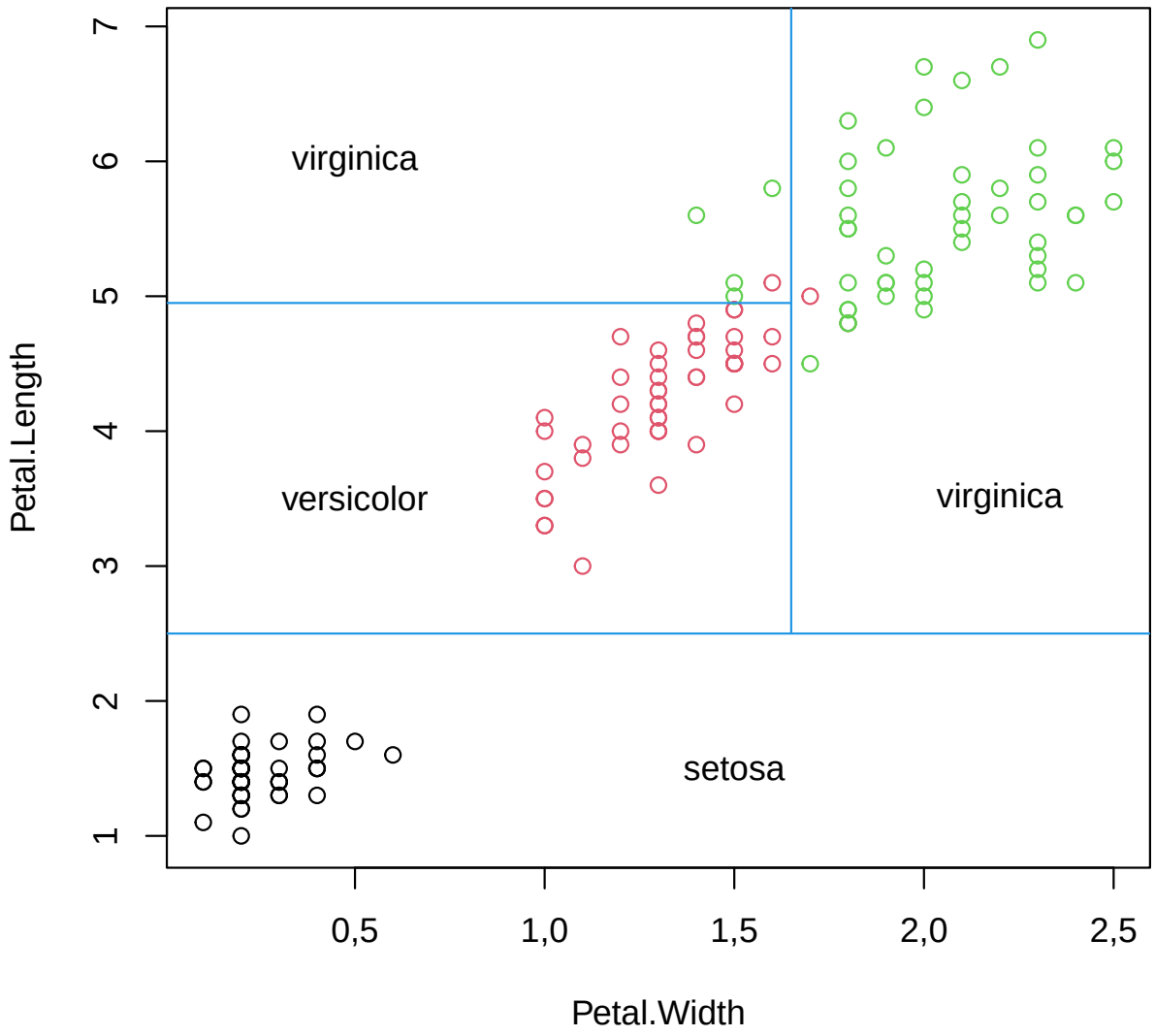

Sepal.Length
4,5
5,0
5,5
6,0
6,5
7,0
7,5
8,0
setosa
versicolor
virginica
Species


Sepal.Width
4,0
3,5
3,0
2,5
2,0
setosa
versicolor
virginica
Species

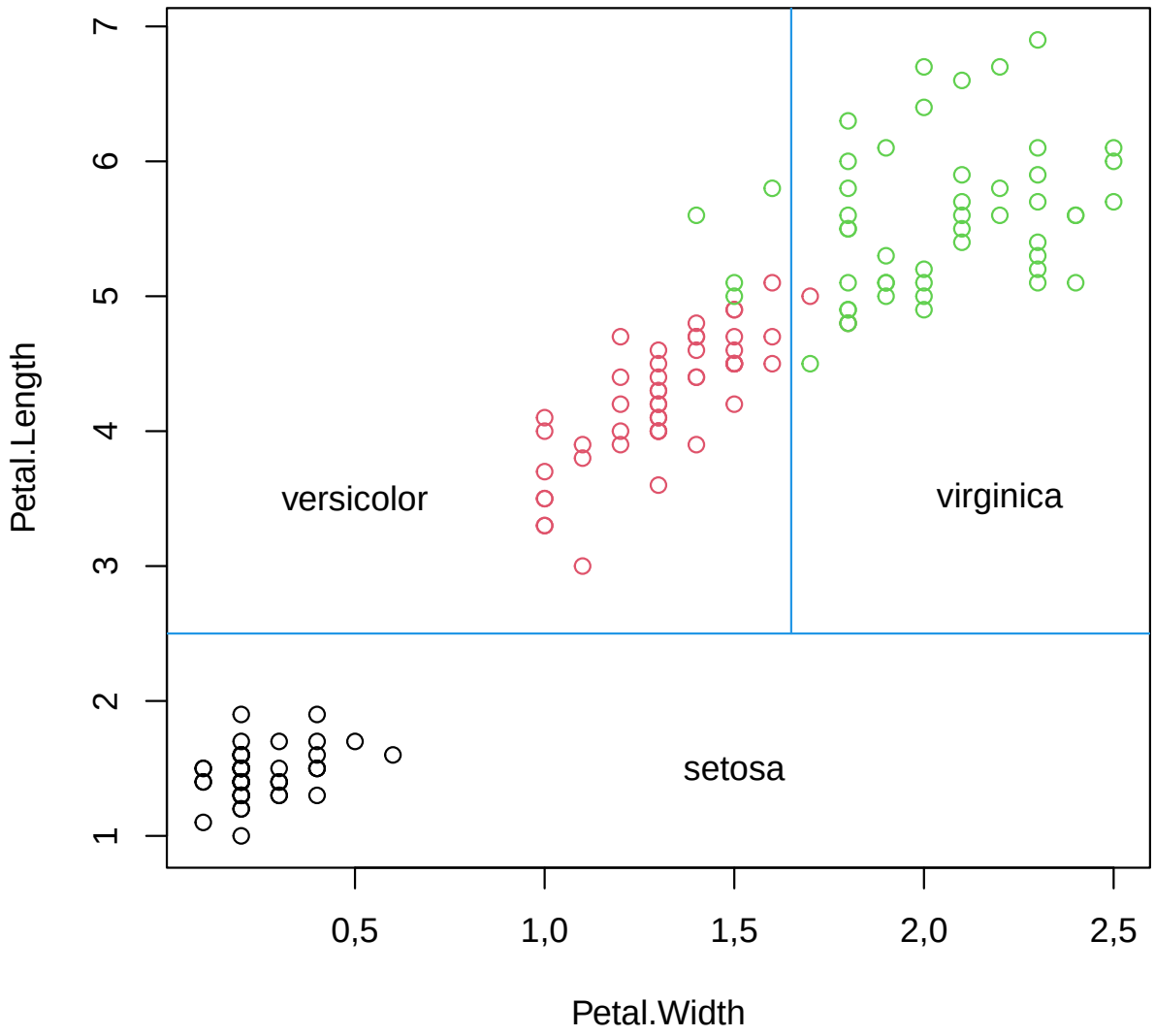

Petal.Length
1
2
3
4
5
6
7
0,5
1,0
1,5
2,0
2,5
Petal.Width
setosa
versicolor
virginica


Petal.Length
Petal.Width
setosa
1
2
3
4
5
6
7
0,5
1,0
1,5
2,0
2,5

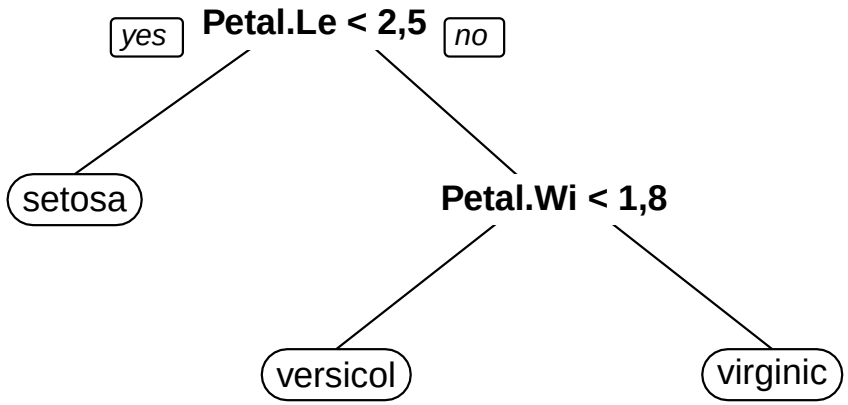

Petal.Length
Petal.Width
versicolor
virginica
setosa
1
2
3
4
5
6
7
0,5
1,0
1,5
2,0
2,5


virginica
versicolor
virginica
setosa
Petal.Length
Petal.Width
1
2
3
4
5
6
7
0,5
1,0
1,5
2,0
2,5

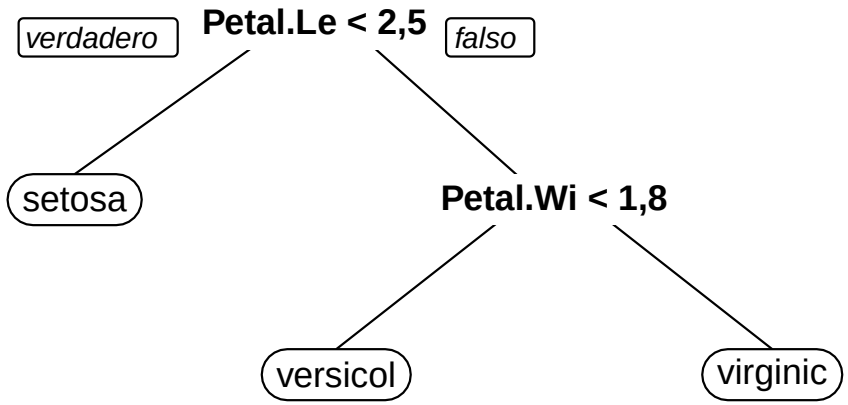
```
> library (rpart.plot) # basta rpart para cálculos
> árbol <- rpart (Species ~
+                   Petal.Length + Petal.Width,
+                 iris)
> árbol
n= 150

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

1) root 150 100 setosa (0,3333 0,3333 0,3333)
  2) Petal.Length< 2.45 50   0 setosa (1,0000 0,0000
  3) Petal.Length>=2.45 100  50 versicolor (0,0000 0,
    6) Petal.Width< 1.75 54   5 versicolor (0,0000 0,
    7) Petal.Width>=1.75 46   1 virginica (0,0000 0,0
```


Petal.Length
Petal.Width
versicolor
virginica
setosa
1
2
3
4
5
6
7
0,5
1,0
1,5
2,0
2,5

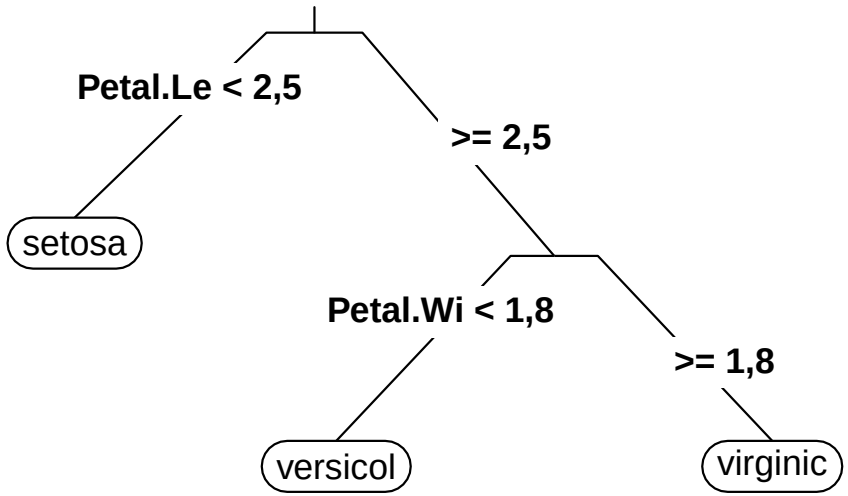
```
> prp (árbol)
```



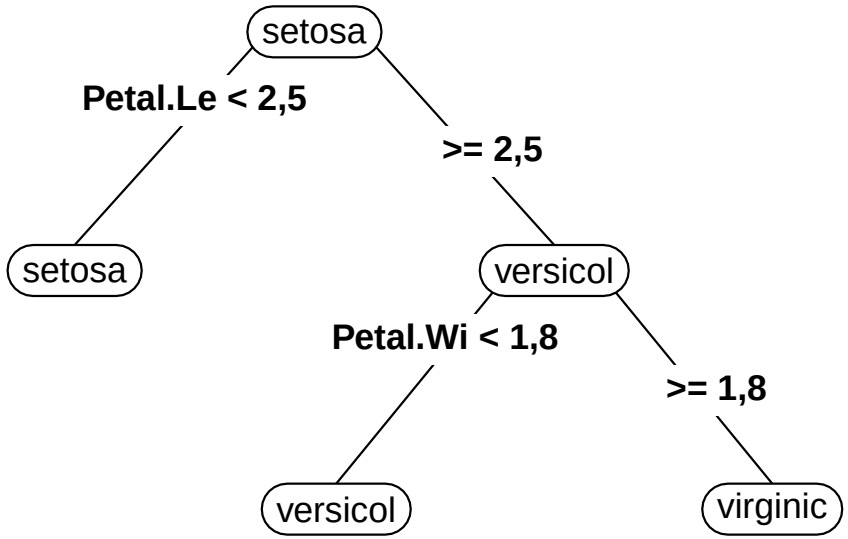
```
> prp (árbol, yes.text="verdadero", no.text="falso")
```



```
> prp (árbol, type=3)
```



```
> prp (árbol, type=4)
```



Nomenclatura

- predictores: $X_1, \ldots, X_p$
- respuesta: Y
  - cualitativa: árbol de clasificación
  - cuantitativa: árbol de regresión
- el árbol consta de *ramas*
- cada rama nace de un *nodo* o *nudo*
- cada nodo representa un subconjunto de la muestra
- la muestra completa es el nodo *raíz*
- nodo $k^{\underline{o}}$: regla de decisión asociada a cierta X_i
  - X_i cualitativa: ¿$X_i \in A_{ik}$?
  - X_i cuantitativa: ¿$X_i \leq c_k$?
- los nodos finales se llaman *hojas*; representan una partición de la muestra

Algoritmos

CHAID 1980. CHi^2 Automatic Interaction Detection. Ramificaciones múltiples. Sólo clasificación.

CART 1984. Classification and Regression Trees. Ramificaciones binarias. El más usado en R.

ID3→C4.5 1986. El más popular en Weka (llamado J48). Ramificaciones múltiples. Sólo clasificación.

MARS 1991. Multivariate Adaptive Regression Splines. Sólo regresión (lineal a trozos).

CIT 2006. Conditional Inference Trees. Basados en contrastes múltiples no paramétricos.

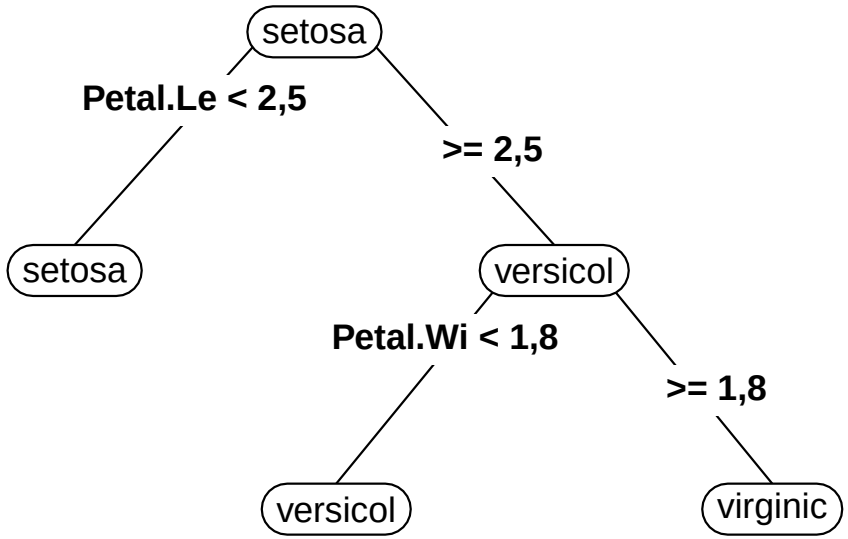
Elementos para la construcción

- Método para elegir hoja candidata a división.
- Criterio de parada.
- Método para asignar a un nodo una predicción.

Notación de los nodos

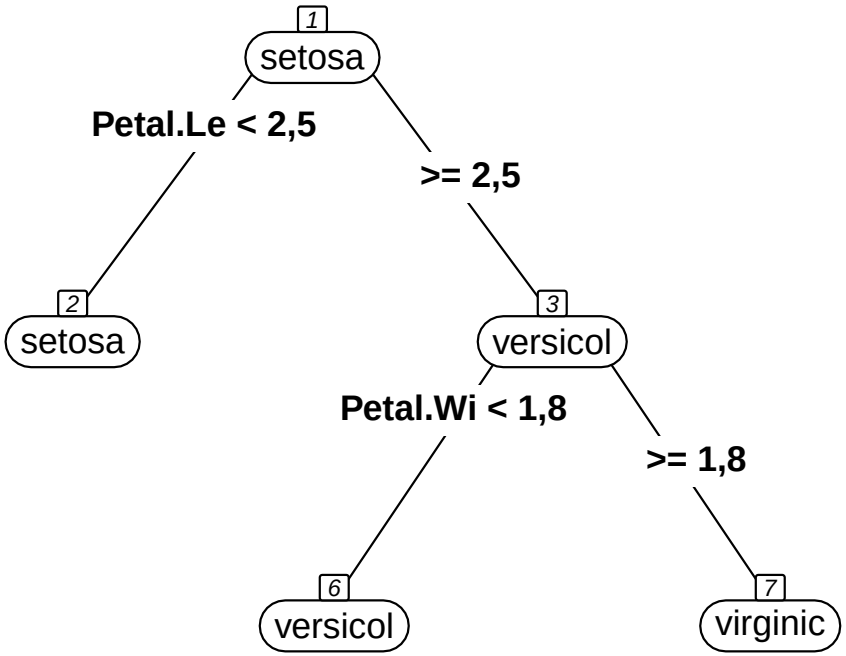
- t un cierto nodo
- t_L nodo hijo izquierdo
- t_R nodo hijo derecho
- T conjunto de las hojas del árbol

```
> prp (árbol, type=4)
```

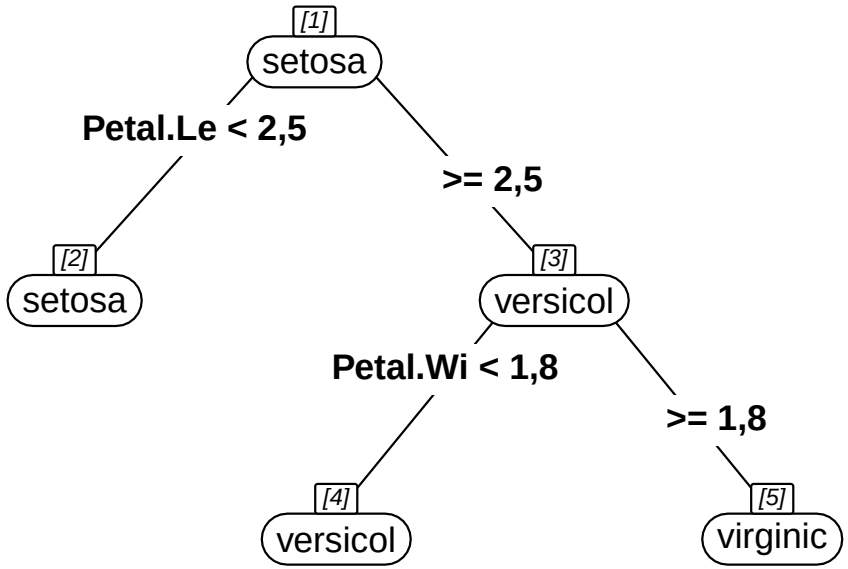


```
> árbol $ frame [, 1:5]          # todos los nodos
           var   n  wt dev yval
1 Petal.Length 150 150 100    1
2      <leaf>  50  50   0    1
3  Petal.Width 100 100  50    2
6      <leaf>  54  54   5    2
7      <leaf>  46  46   1    3
> árbol $ where [c(1:5,145:150)] # hojas
  1   2   3   4   5 145 146 147 148 149 150
  2   2   2   2   2   5   5   5   5   5   5
> table (árbol$where)
 2  4  5
50 54 46
```

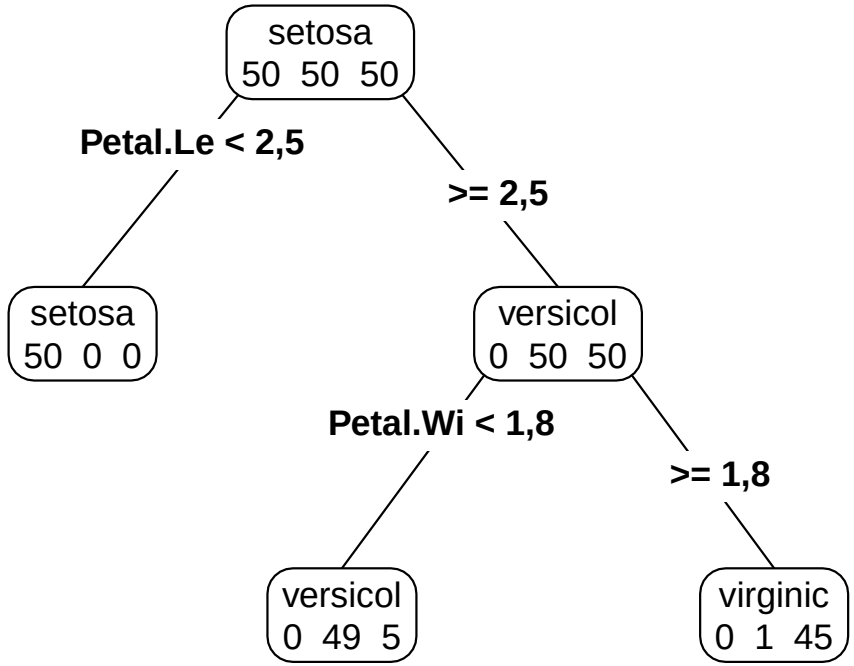
```
> prp (árbol, type=4, nn=TRUE)
```



```
> prp (árbol, type=4, ni=TRUE)
```



```
> prp (árbol, type=4, extra=1)
```



Elegir siguiente corte: medidas de impureza

$$\begin{array}{rccc} \Phi: & \mathbb{P} & \to & \mathbb{R} \\ & (p_1, \ldots, p_k) & \mapsto & \Phi(p_1, \ldots, p_k) \end{array}$$

donde

$$\mathbb{P} = \{(p_1, \ldots, p_k) \in [0;1]^k \mid p_1 + \cdots + p_k = 1\}$$

Requisitos:

- Alcanzar máximo en $(\frac{1}{k}, \ldots, \frac{1}{k})$.
- Alcanzar mínimo en conjuntos homogéneos como $(0, \ldots, 0, 1, 0, \ldots 0)$.
- Invariante frente a permutaciones de $(p_1, \ldots, p_k)$.

- impureza del nodo t:

$$i(t) = \Phi(\Pr[1 \mid t], \ldots, \Pr[k \mid t])$$

donde $\Pr[j \mid t]$ es la probabilidad de la clase j en el nodo t

- reducción de la impureza al dividir t en $\{t_L, t_R\}$

$$\Delta i(t \rightarrow t_L, t_R) = i(t) - p_L i(t_L) - p_R i(t_R)$$

donde

  - p_L es la proporción de t que se va a t_L
  - p_R es la proporción de t que se va a $t_R = 1 - p_L$

- impureza del árbol

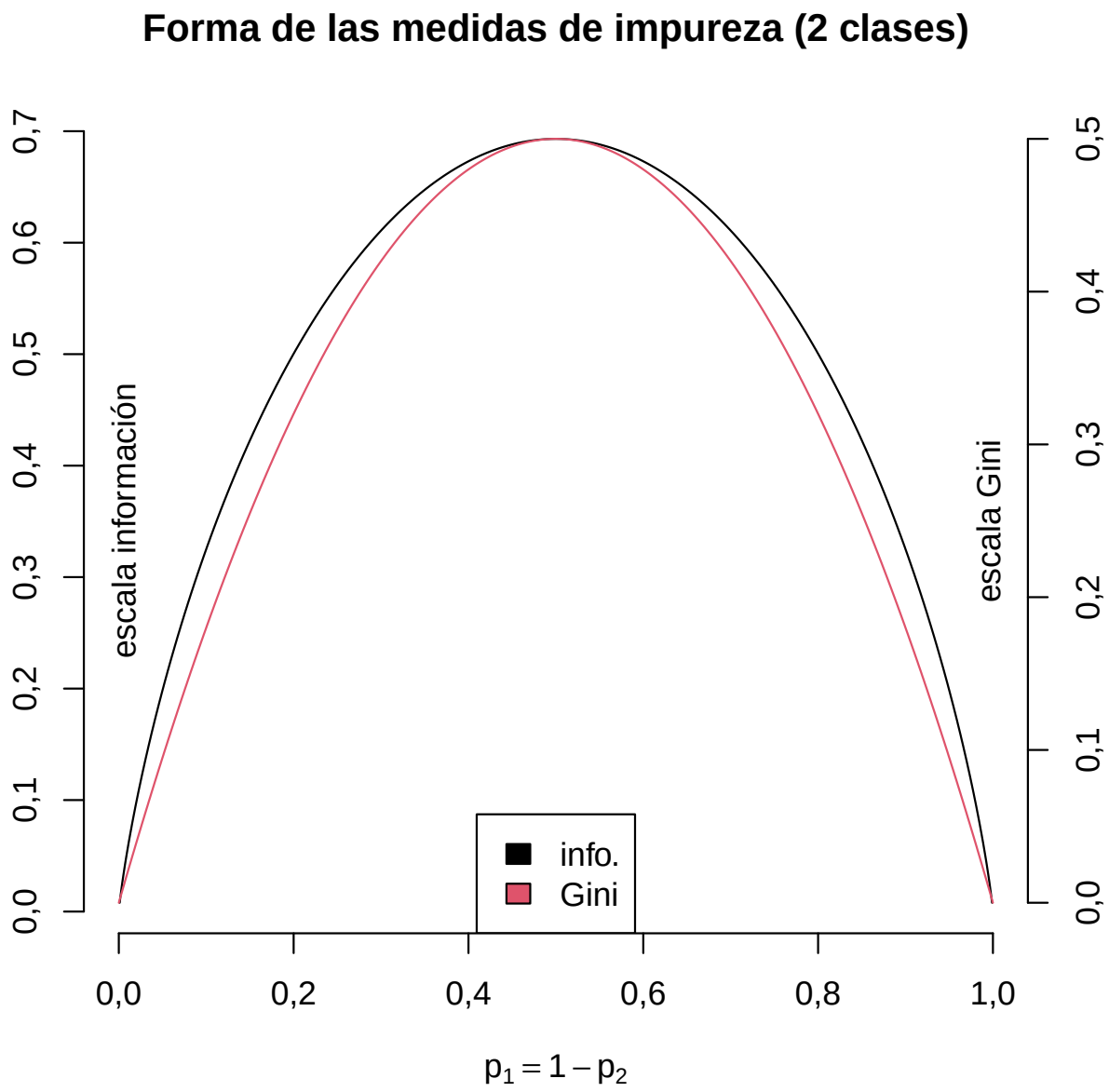
$$I(T) = \sum_{t \in T} \Pr(t)\, i(t)$$

- medidas habituales
  - entropía o información

$$-\sum_{j} p_j \log p_j$$

  - Gini

$$\sum_{j} p_j (1 - p_j) = 1 - \sum_{j} p_j^2$$


Forma de las medidas de impureza (2 clases)
escala información
escala Gini
0,0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,0
0,1
0,2
0,3
0,4
0,5
info.
Gini
0,0
0,2
0,4
0,6
0,8
1,0
$p_1 = 1 - p_2$

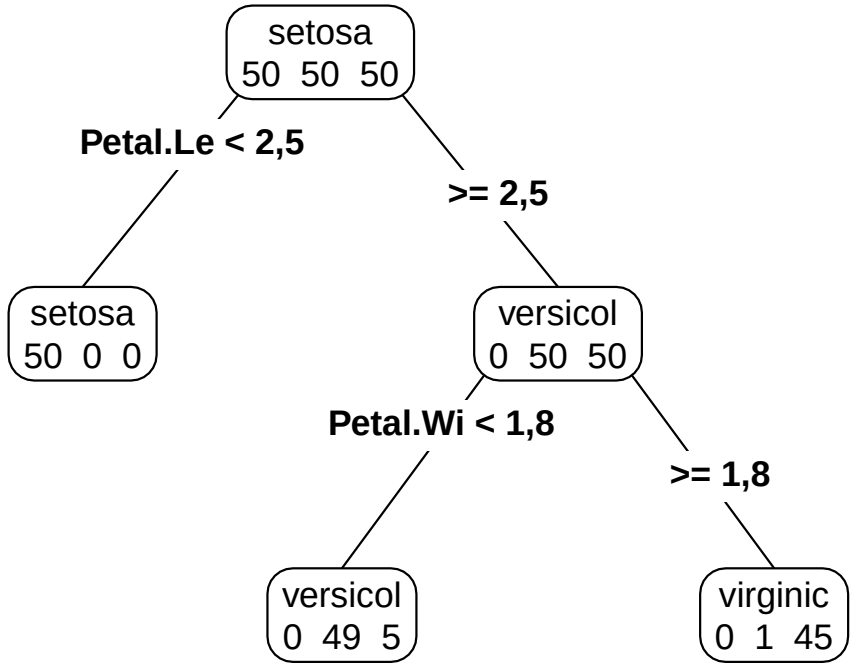
```
> ## por omisión se usa Gini
> a.gini <- rpart (Species ~
+                    Petal.Length + Petal.Width,
+                  iris,
+                  parms = list(split="gini"))
> a.info <- rpart (Species ~
+                    Petal.Length + Petal.Width,
+                  iris,
+                  parms = list(split="information"))
```

Riesgo

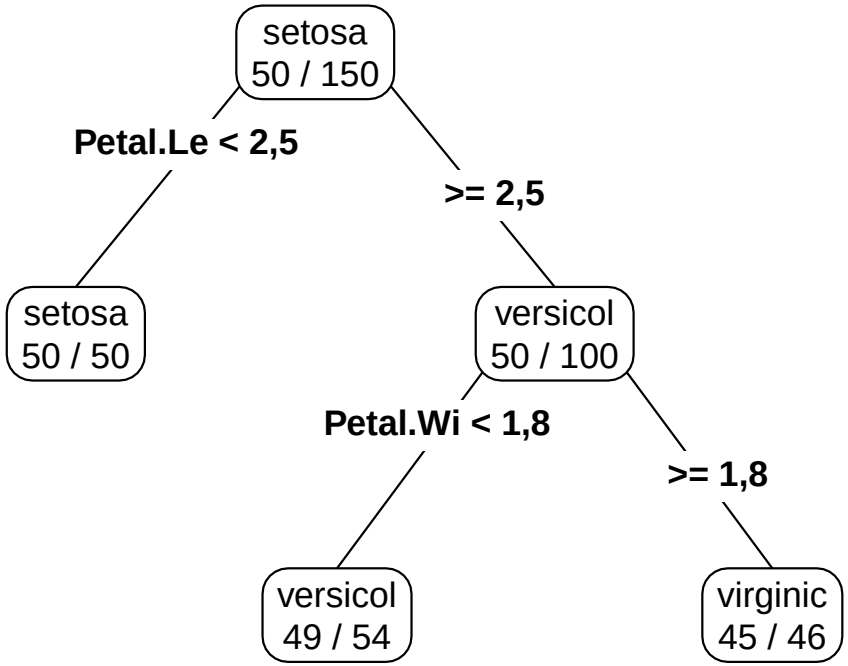
- riesgo de una hoja

$$R(t) = \Pr[\text{error clasif.} \mid t] = 1 - \max_{j \in \text{clases}} \Pr[j \mid t]$$

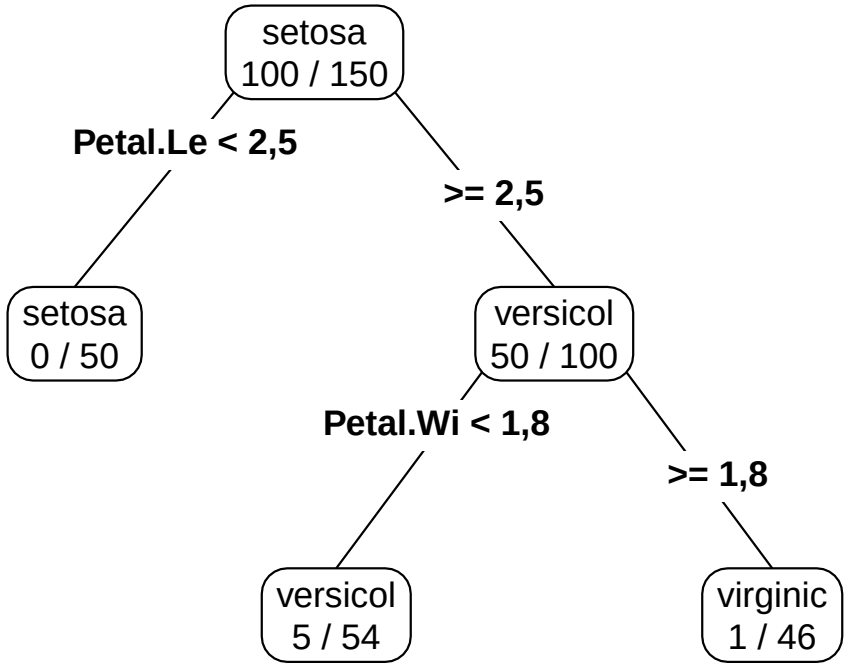
```
> prp (árbol, type=4, extra=1)
```



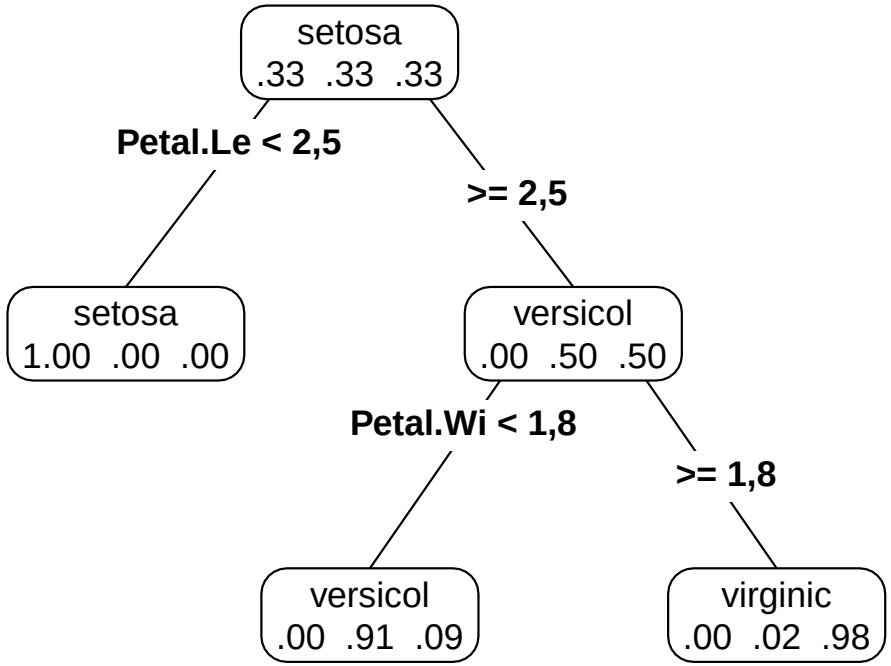
```
> prp (árbol, type=4, extra=2)
```



```
> prp (árbol, type=4, extra=3)
```



```
> prp (árbol, type=4, extra=4)
```



Riesgo

- riesgo de una hoja

$$R(t) = \Pr[\text{error clasif.} \mid t] = 1 - \max_{j \in \text{clases}} \Pr[j \mid t]$$

- riesgo del árbol

$$R(T) = \Pr[\text{error clasif.}] = \sum_{t \in T} R(t) \Pr[t]$$

- disminución del riesgo si $t \to \{t_L, t_R\}$

$$\Delta r = R(t) - R(t_L, t_R) > 0$$

- posible criterio de corte: maximizar Δr

¿Riesgo como criterio de corte?

- ¿maximizar Δr?
- supóngase:
  - 80 % de individuos de clase 1 en la raíz
  - cierto corte candidato llevaría a
    - 54 % de clase 1 en $t_L \Rightarrow$ asignar clase 1
    - 100 % de clase 1 en $t_R \Rightarrow$ asignar clase 1

  pero $\Delta r = 0$ aunque la bifurcación es muy informativa
- supónganse dos cortes candidatos:
  - corte A llevaría a 70 % y 70 % de clase 1
  - corte B llevaría a 85 % y 50 % de clase 1
  - el A tiene menos riesgo
  - el B es preferible en la práctica, porque establece mejor cómo seguir dividiendo

Criterios de parada

minsplit Tamaño mínimo antes de cortar.
Por omisión, 20.

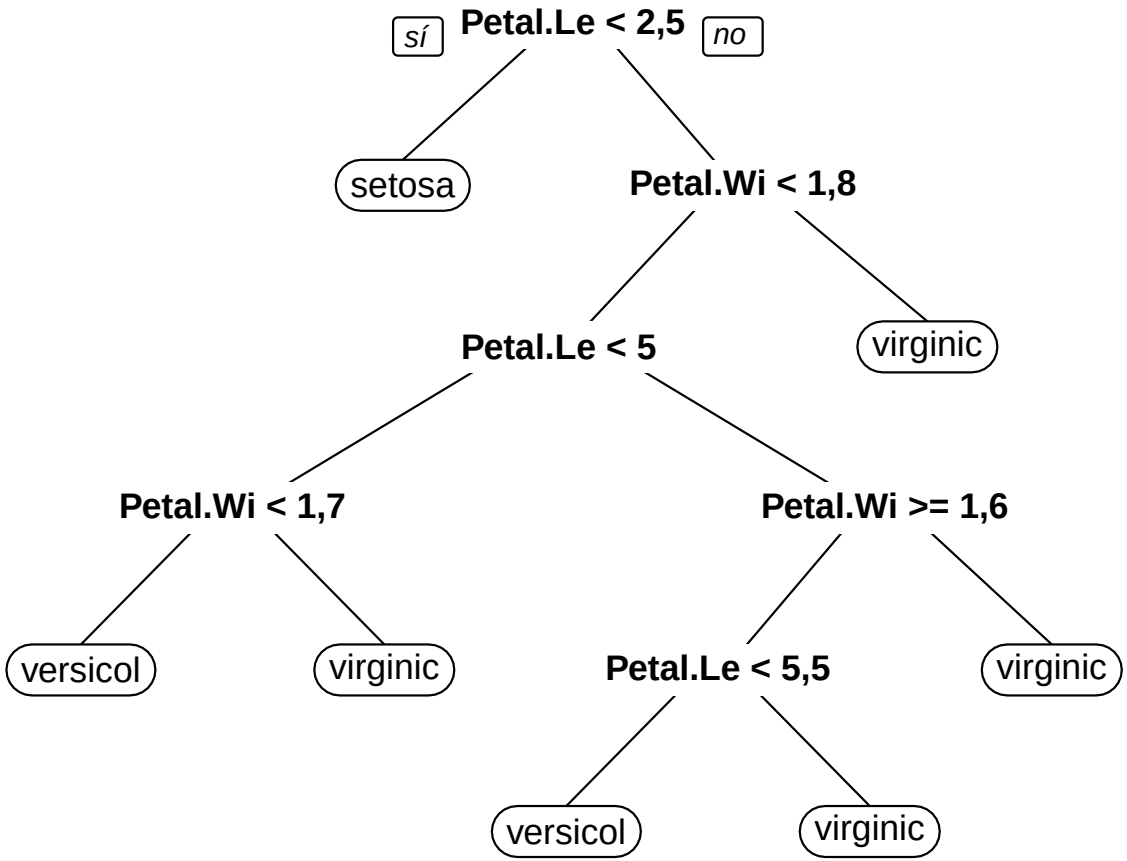
minbucket Tamaño mínimo de cada hoja. Por omisión, 7.

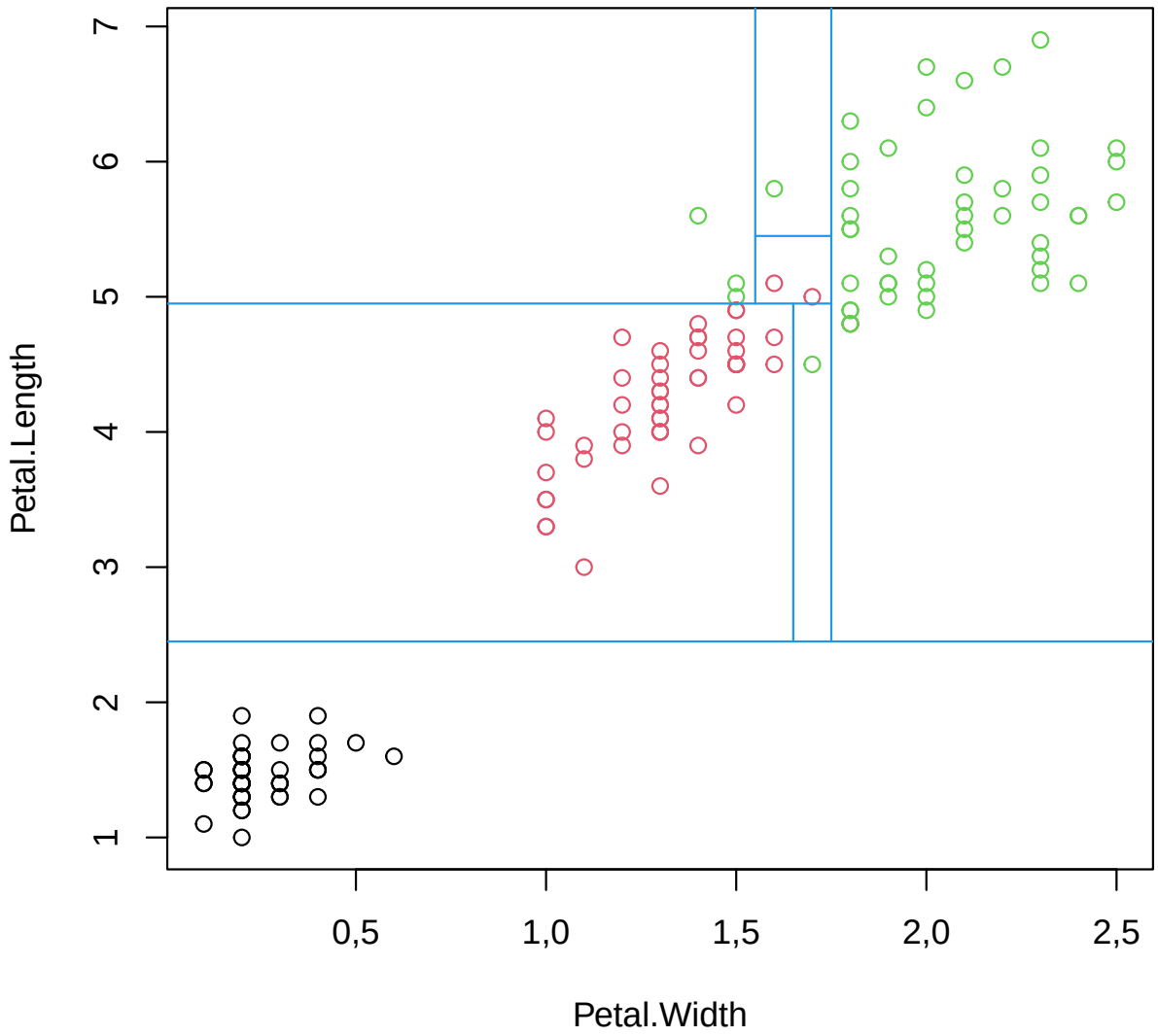
cp Parámetro de complejidad.
Proporción mínima de mejora en el ajuste.
Por omisión, 1 %.

maxdepth Máxima profundidad de las hojas
(0 = profundidad de la raíz).
Por omisión, 30.

```
> árbol$frame[,1:6]
          var   n  wt dev yval complexity
1 Petal.Length 150 150 100    1       0,50
2       <leaf>  50  50   0    1       0,01
3  Petal.Width 100 100  50    2       0,44
6       <leaf>  54  54   5    2       0,00
7       <leaf>  46  46   1    3       0,01
> table (iris$Species,
+        predict (árbol, iris, type="class"))
              setosa versicolor virginica
  setosa          50          0         0
  versicolor       0         49         1
  virginica        0          5        45
```

```
> rpart (Species~Petal.Length+Petal.Width, iris,
+    control = rpart.control (cp = 0, minsplit = 2))
n= 150

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

 1) root 150 100 setosa (0,3333 0,3333 0,3333)
   2) Petal.Length< 2.45 50   0 setosa (1,0000 0,0000
   3) Petal.Length>=2.45 100  50 versicolor (0,0000 0
     6) Petal.Width< 1.75 54   5 versicolor (0,0000 0
      12) Petal.Length< 4.95 48   1 versicolor (0,000
        24) Petal.Width< 1.65 47   0 versicolor (0,00
        25) Petal.Width>=1.65 1   0 virginica (0,0000
      13) Petal.Length>=4.95 6   2 virginica (0,0000
        26) Petal.Width>=1.55 3   1 versicolor (0,000
          52) Petal.Length< 5.45 2   0 versicolor (0,
          53) Petal.Length>=5.45 1   0 virginica (0,0
        27) Petal.Width< 1.55 3   0 virginica (0,0000
     7) Petal.Width>=1.75 46   1 virginica (0,0000 0,
```


sí
Petal.Le < 2,5
no
setosa
Petal.Wi < 1,8
Petal.Le < 5
virginic
Petal.Wi < 1,7
Petal.Wi >= 1,6
versicol
virginic
Petal.Le < 5,5
virginic
versicol
virginic


Petal.Length
1
2
3
4
5
6
7
0,5
1,0
1,5
2,0
2,5
Petal.Width

Criterio de poda

- T_∞ árbol sin ramas, sólo raíz
- $\alpha > 0$ parámetro de complejidad
- $R_\alpha(T) = R(T) + \alpha \cdot |T| \cdot R(T_\infty)$
- T_α único árbol que minimiza R_α
- T_0 árbol completo

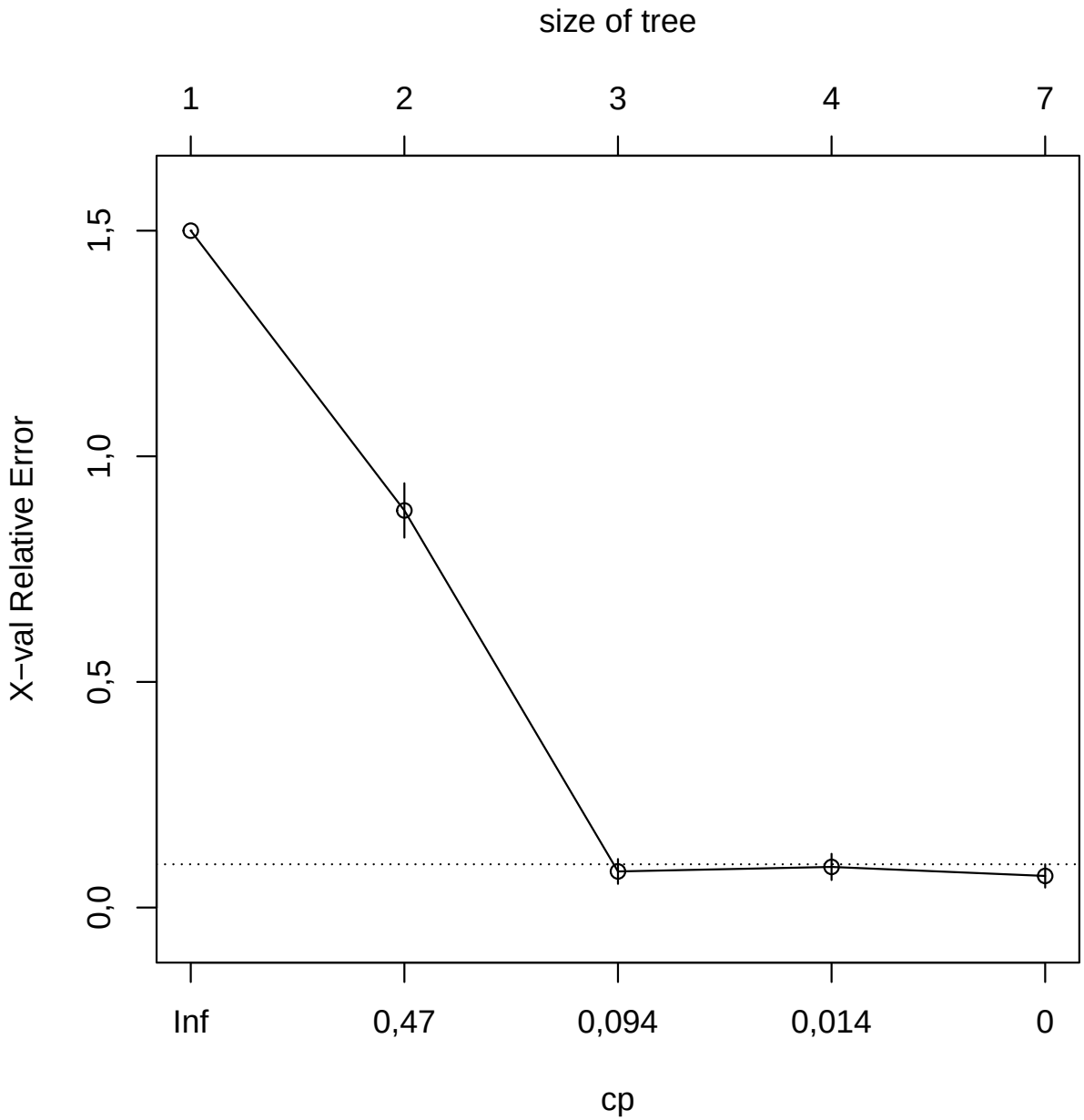
```
> printcp (árbol)                    # árbol$cptable
Classification tree:
rpart(formula = Species ~ Petal.Length + Petal.Width,

Variables actually used in tree construction:
[1] Petal.Length Petal.Width

Root node error: 100/150 = 0,7

n= 150

    CP nsplit rel error xerror xstd
1 0,50      0      1,00   1,13 0,05
2 0,44      1      0,50   0,62 0,06
3 0,01      2      0,06   0,09 0,03
```

Validación cruzada

Cambiar número de cruzvalidaciones para obtener resultados más estables:

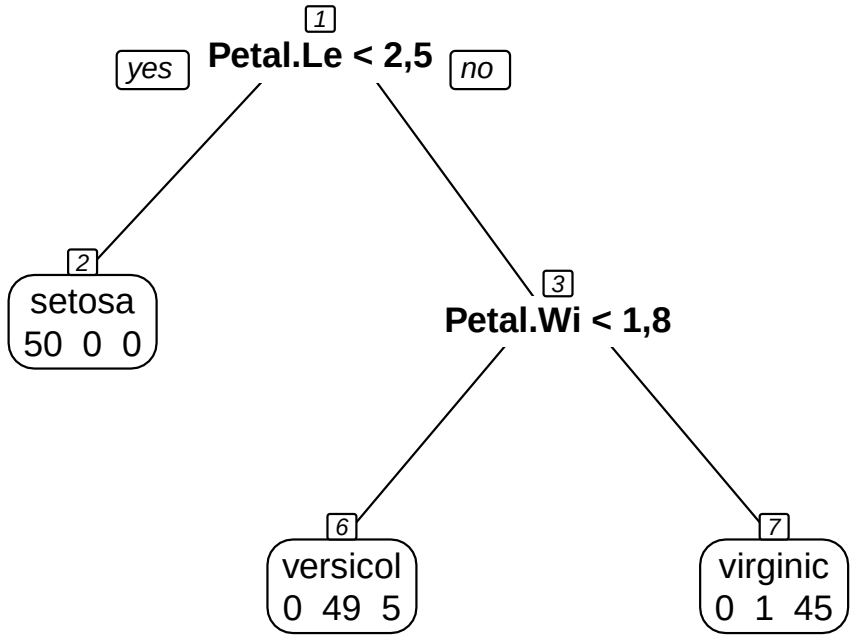
```
> a100 <- rpart (Species~Petal.Length+Petal.Width,
+                iris,
+                control=rpart.control(cp=0,
+                                      minsplit=2,
+                                      xval=100))
> a100$cptable
    CP nsplit rel error xerror   xstd
1 0,50      0      1,00   1,50 0,0000
2 0,44      1      0,50   0,88 0,0603
3 0,02      2      0,06   0,08 0,0275
4 0,01      3      0,04   0,09 0,0291
5 0,00      6      0,01   0,07 0,0258
```

```
> plotcp (a100) # abscisas = medias geométricas
```



```
> prune (a100, cp=0.09) # cualq. entre 0,02 y 0,44
n= 150

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

1) root 150 100 setosa (0,3333 0,3333 0,3333)
  2) Petal.Length< 2.45 50   0 setosa (1,0000 0,0000
  3) Petal.Length>=2.45 100  50 versicolor (0,0000 0,
    6) Petal.Width< 1.75 54   5 versicolor (0,0000 0,
    7) Petal.Width>=1.75 46   1 virginica (0,0000 0,0
```

Importancia de las variables



Importancia de las variables

- suma de calidad de la división en nodos cuyas reglas involucran la variable
- en la salida de `summary` aparece en porcentajes
- si dos variables están muy correladas, su importancia conjunta se dividirá entre las dos y puede que ninguna destaque

```
> árbol $ variable.importance
 Petal.Width Petal.Length
        89,0         81,3
```

Importancia de las variables

```
> summary (árbol)
Call:
rpart(formula = Species ~ Petal.Length + Petal.Width,
  n= 150

    CP nsplit rel error xerror   xstd
1 0,50      0      1,00   1,13 0,0528
2 0,44      1      0,50   0,62 0,0603
3 0,01      2      0,06   0,09 0,0291

Variable importance
 Petal.Width Petal.Length
          52           48

Node number 1: 150 observations,    complexity param=
  predicted class=setosa       expected loss=0.667  P(
    class counts:    50    50    50
```

Regresión

- Y cuantitativa
- `rpart` usa el método `anova`
- criterio: maximizar $\mathrm{SC}_t - \mathrm{SC}_L - \mathrm{SC}_R$ donde
  - SC_t = suma de cuadrados $\sum (y_i - \bar{y})^2$ en el nodo t
  - SC_L = ídem en su nodo hijo L
  - SC_R = ídem en su nodo hijo R
- predicción $\hat{y}_t$: al nodo t se le asigna $\bar{y} = \frac{1}{n_t} \sum_{i \in t} y_i$
  (en clasificación asignábamos la clase modal)
- riesgo en el nodo t: varianza $\frac{1}{n_t} \sum_{i \in t} (y_i - \hat{y}_t)^2$
  (en clasificación era la tasa de incorrectos)

Protocolo para construir un árbol

1. Determinar el número adecuado de permutaciones para validación cruzada: ¿`xval` = n, 10...?
2. Crear un árbol completo
   - `cp` = 0
   - `minsplit` = 2 ó `minbucket` = 1
   - ¿`xval`?
3. Fijarse en si `xstd` (errores típicos obtenidos por cruzvalidación) son reducidos. Si no, aumentar `xval`
4. Buscar el parámetro de complejidad (`cp`) adecuado

```
plotcp (arbol)          # o printcp(arbol)
```

5. Podar el árbol (o recalcularlo con `cp`)

```
arbol <- prune (arbol, cp=...)
rpart (..., control=rpart.control(cp=...))
```

Ejercicio 1: iris

- Construye un árbol de clasificación para los datos `iris` a partir de todas sus variables (en la presentación usamos sólo las de los pétalos).
- ¿Se gana algo respecto a usar sólo los pétalos?
- Compáralo con el análisis discriminante.

Ejercicio 2: mtcars

- Construir un modelo de regresión lineal para estimar `mpg` a partir del resto de variables.
- Construir un árbol de regresión con el mismo objetivo.
- Comparar ambos modelos:
  - Con la información que producen las funciones de R.
  - Con el método de retención (entrenamiento y validación).
  - Con validación cruzada.

Ejercicio 3: solder

- Construye un árbol a partir de los datos `solder` para buscar variables que afecten al número de proyecciones (`skips`).

Ejercicio 4: genotipos

- Construye un árbol para los datos `http://bellman.ciencias.uniovi.es/~carleos/master/manadine/curso1/AnalisisDatos1/3-arboles/dat/geno.csv` para predecir `genotipo`.

Ejercicio 5: tejido mamario

- Construye un árbol de clasificación para los datos sobre tejido mamario de la Hoja 2. Compáralo con el resultado obtenido con análisis discriminante.

Más detalles

- https://es.wikipedia.org/wiki/Aprendizaje_basado_en_%C3%A1rboles_de_decisi%C3%B3n
- https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf
- https://www.researchgate.net/publication/263671703_Fifty_Years_of_Classification_and_Regression_Trees