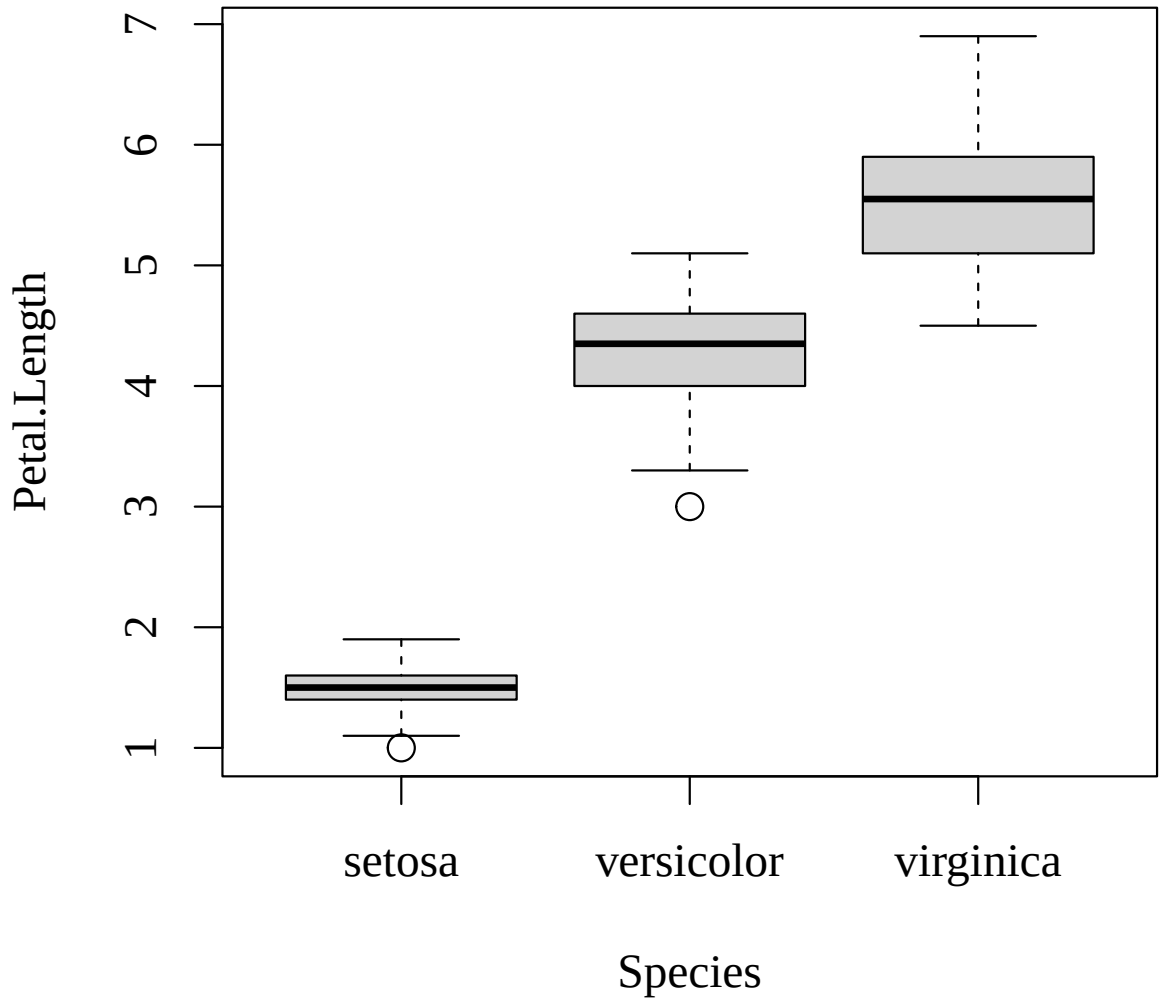
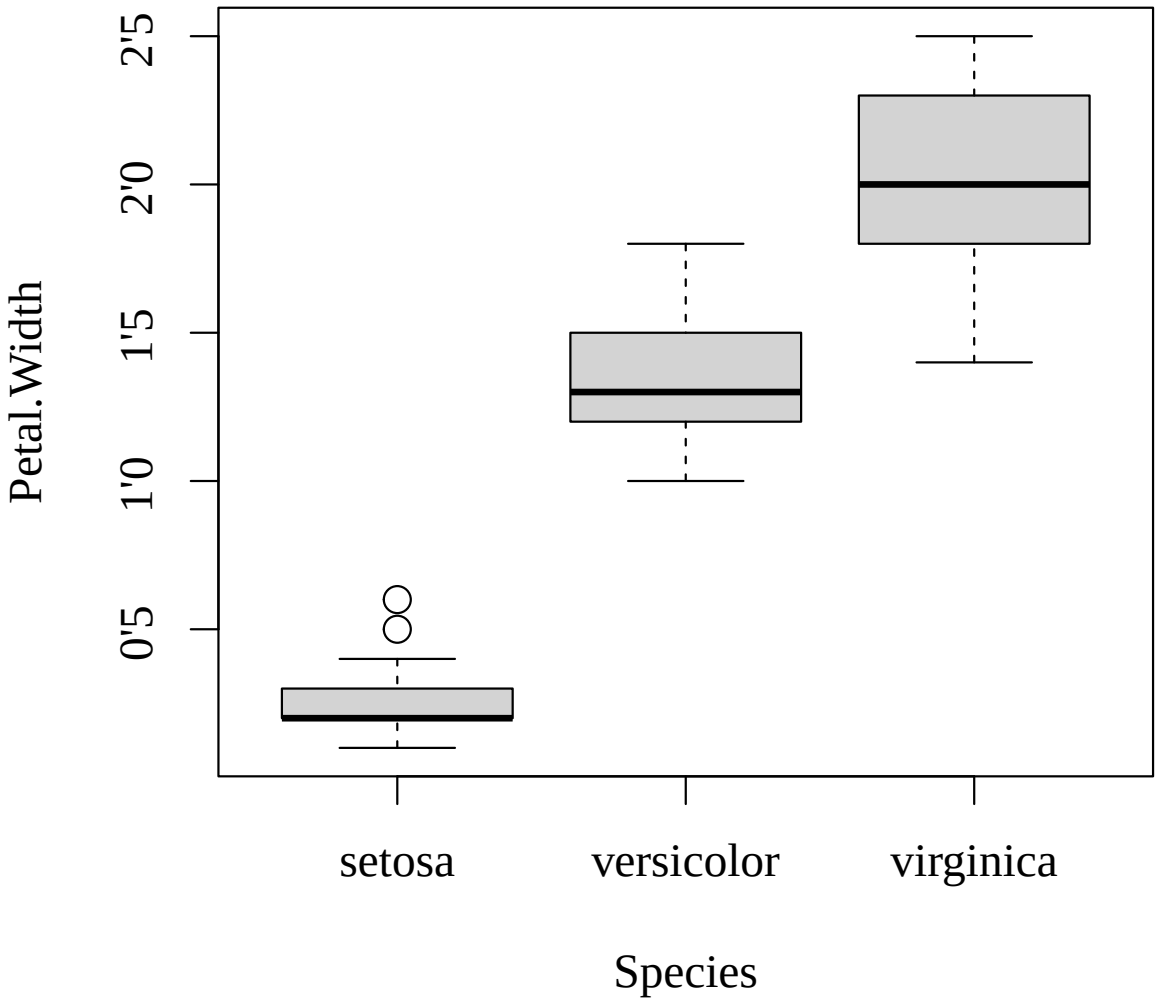
Árboles de decisión

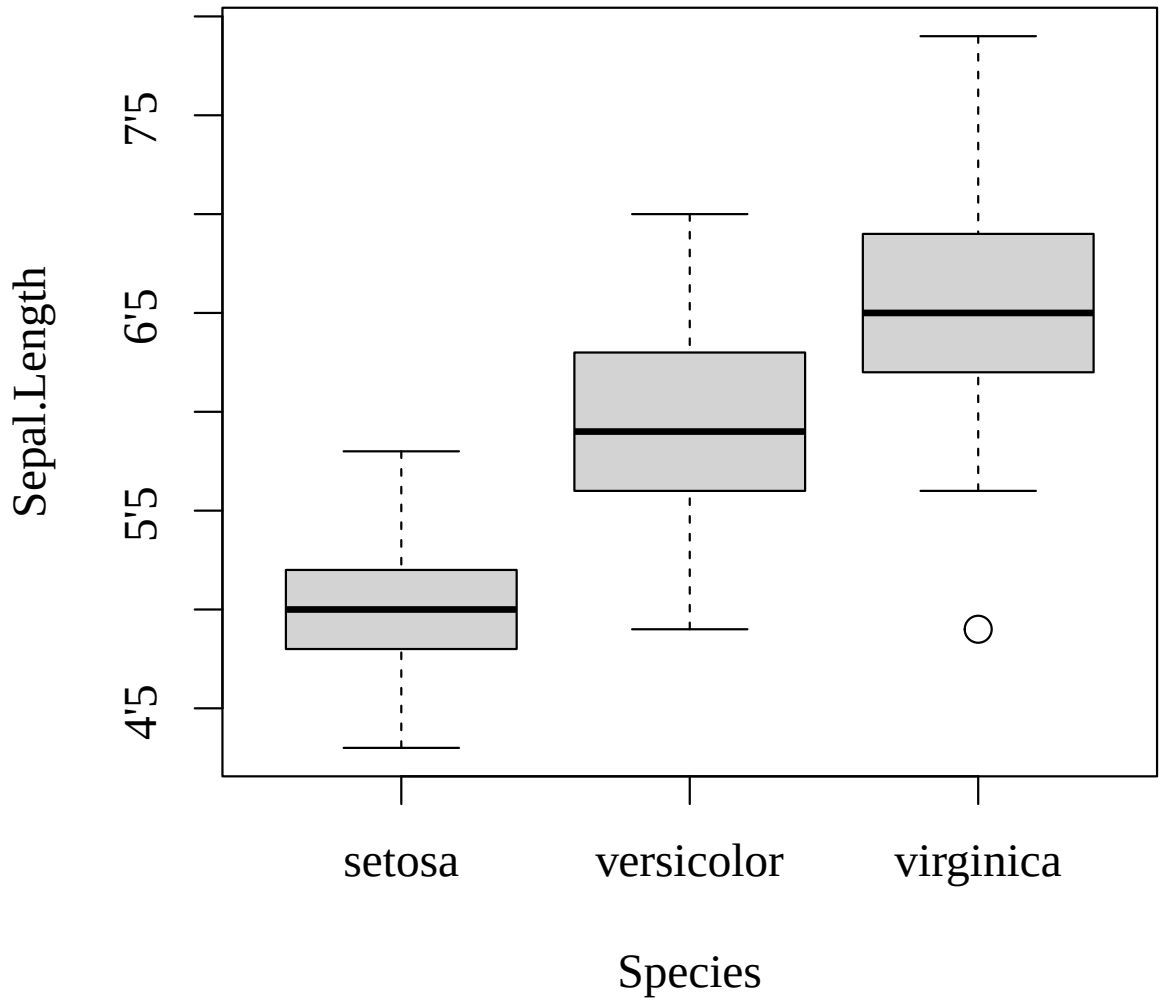
N. Corral,
B. Sinova,
C. Carleos

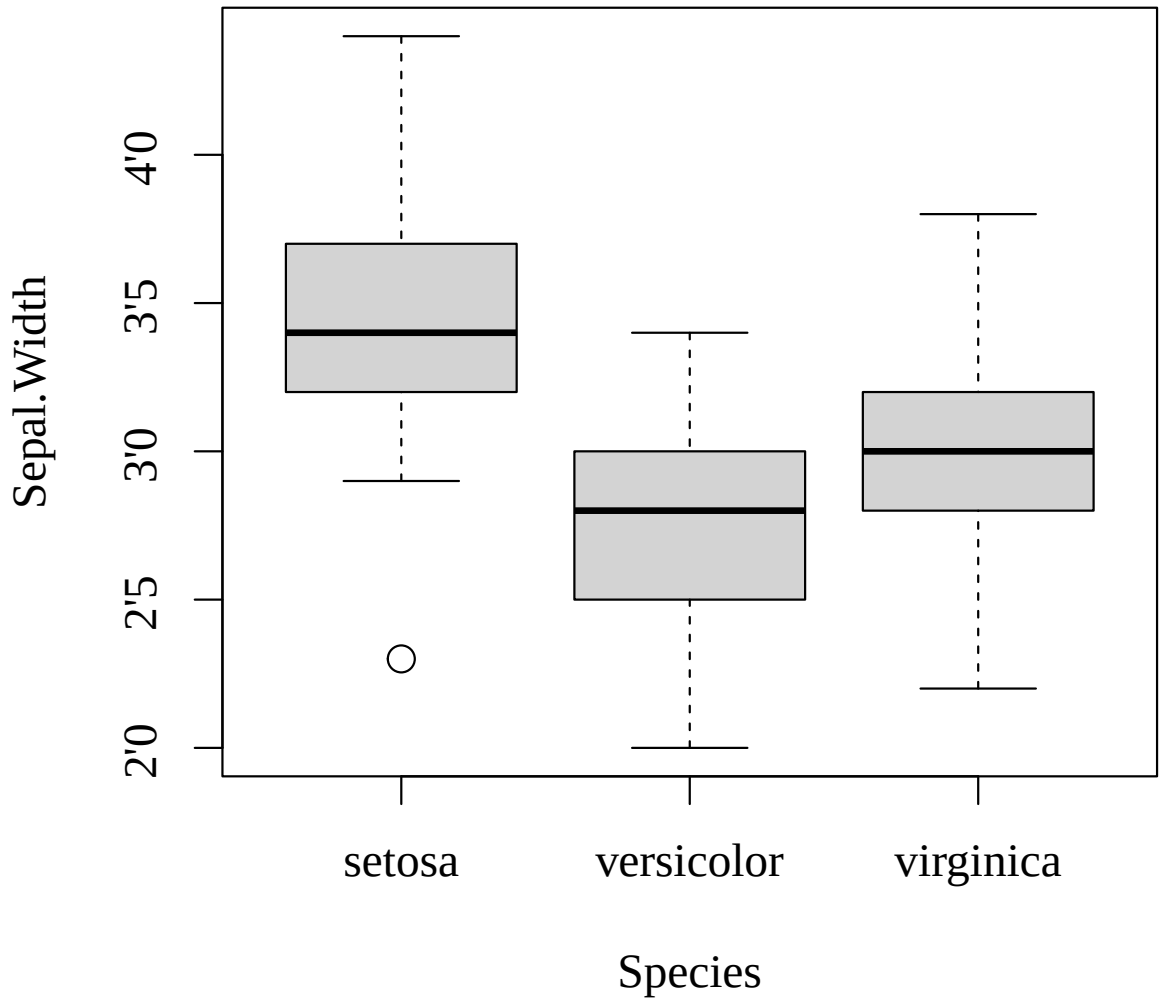
16 de abril de 2026

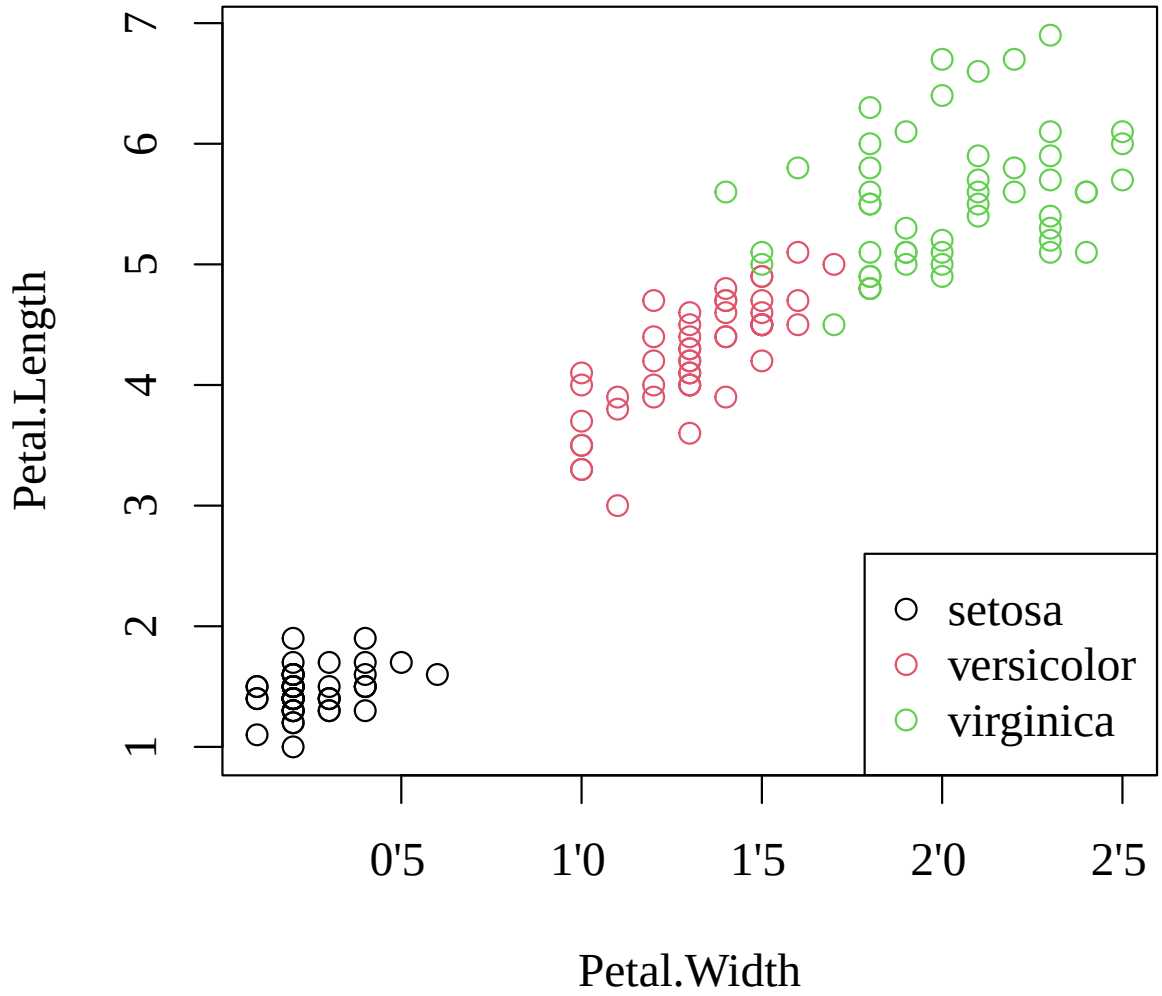
```
> summary (iris)
  Sepal.Length    Sepal.Width     Petal.Length
 Min.   :4'30   Min.   :2'00   Min.   :1'00
 1st Qu.:5'10   1st Qu.:2'80   1st Qu.:1'60
 Median :5'80   Median :3'00   Median :4'35
 Mean   :5'84   Mean   :3'06   Mean   :3'76
 3rd Qu.:6'40   3rd Qu.:3'30   3rd Qu.:5'10
 Max.   :7'90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

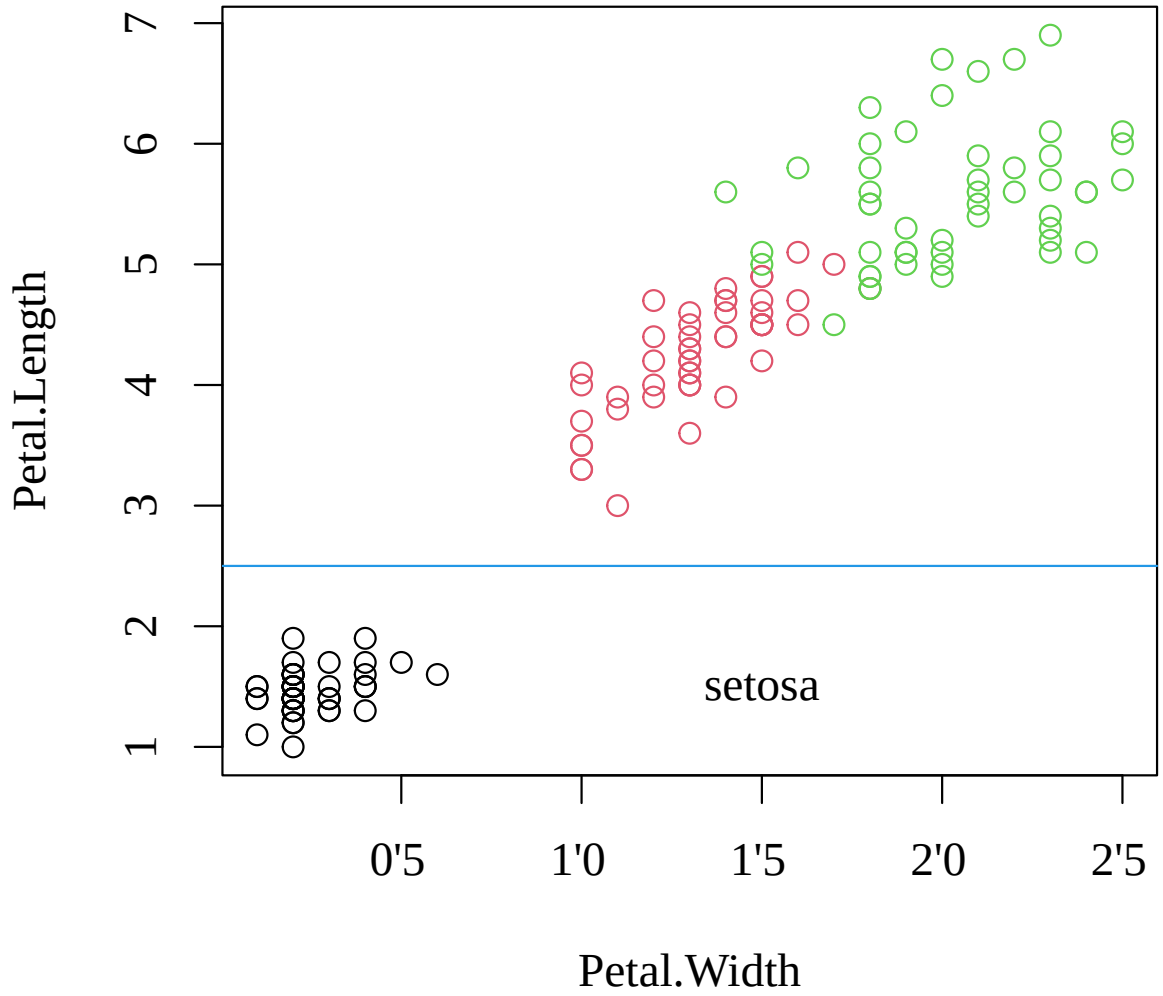


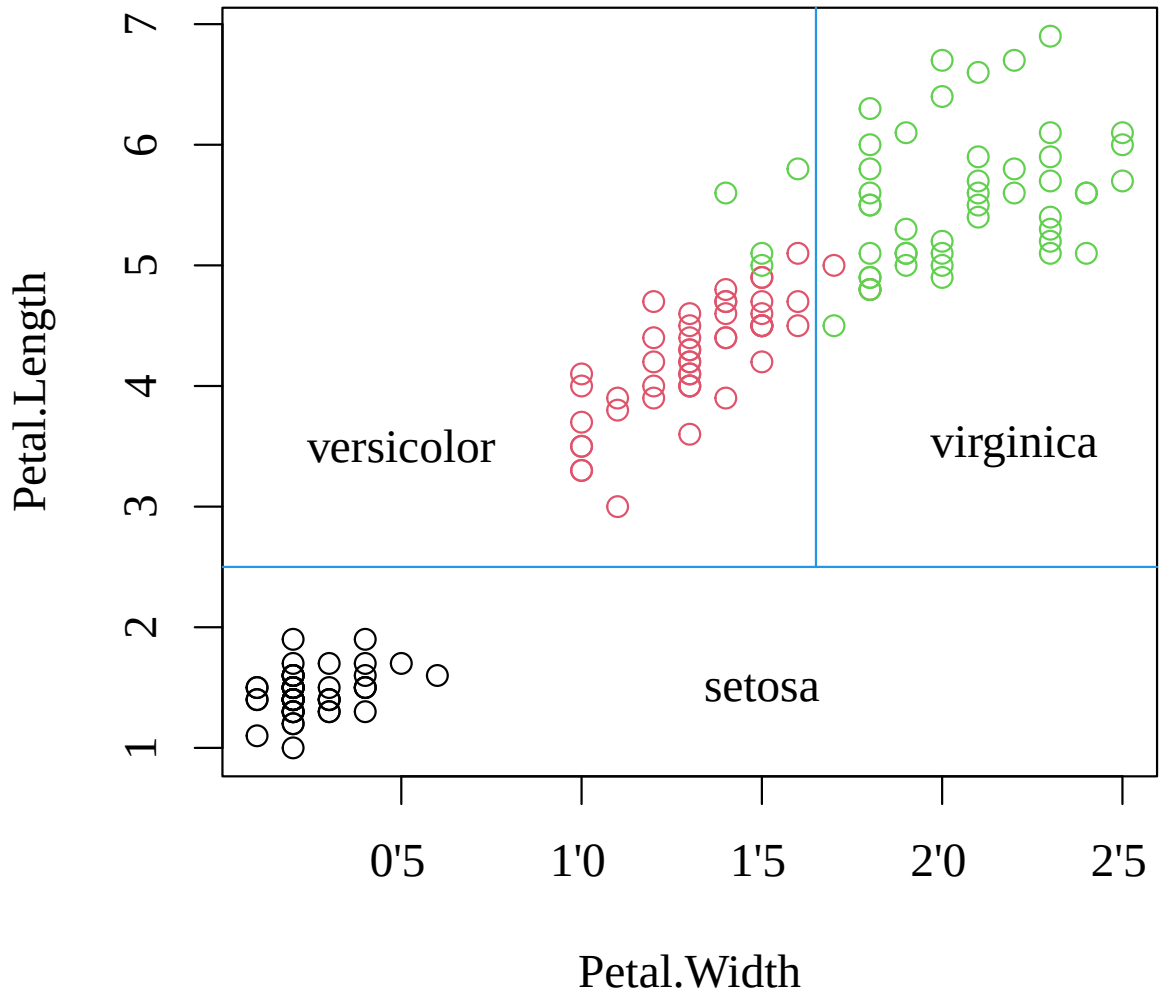

Petal.Width
0'5
1'0
1'5
2'0
2'5
setosa
versicolor
virginica
Species

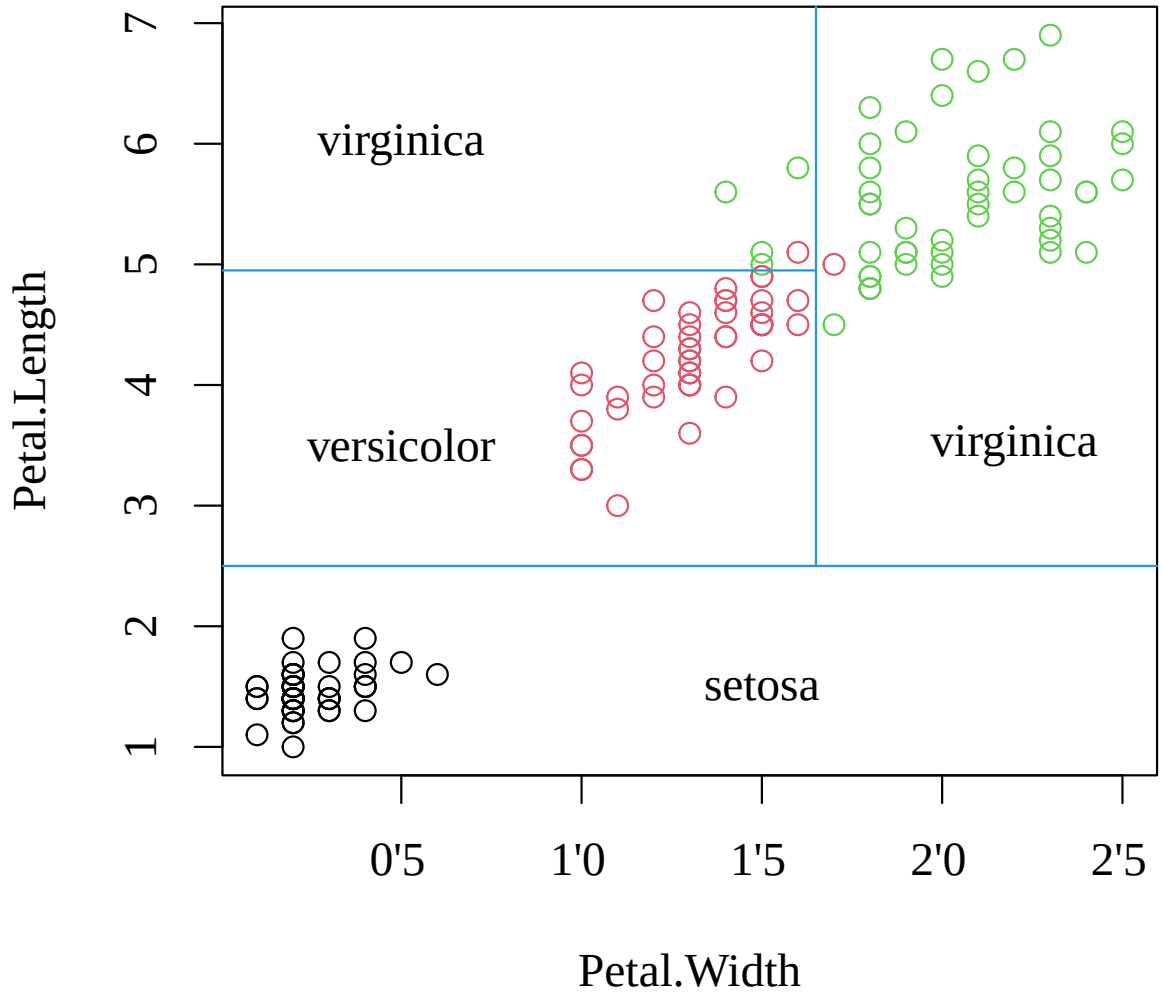

Sepal.Length
4'5
5'5
6'5
7'5
setosa
versicolor
virginica
Species



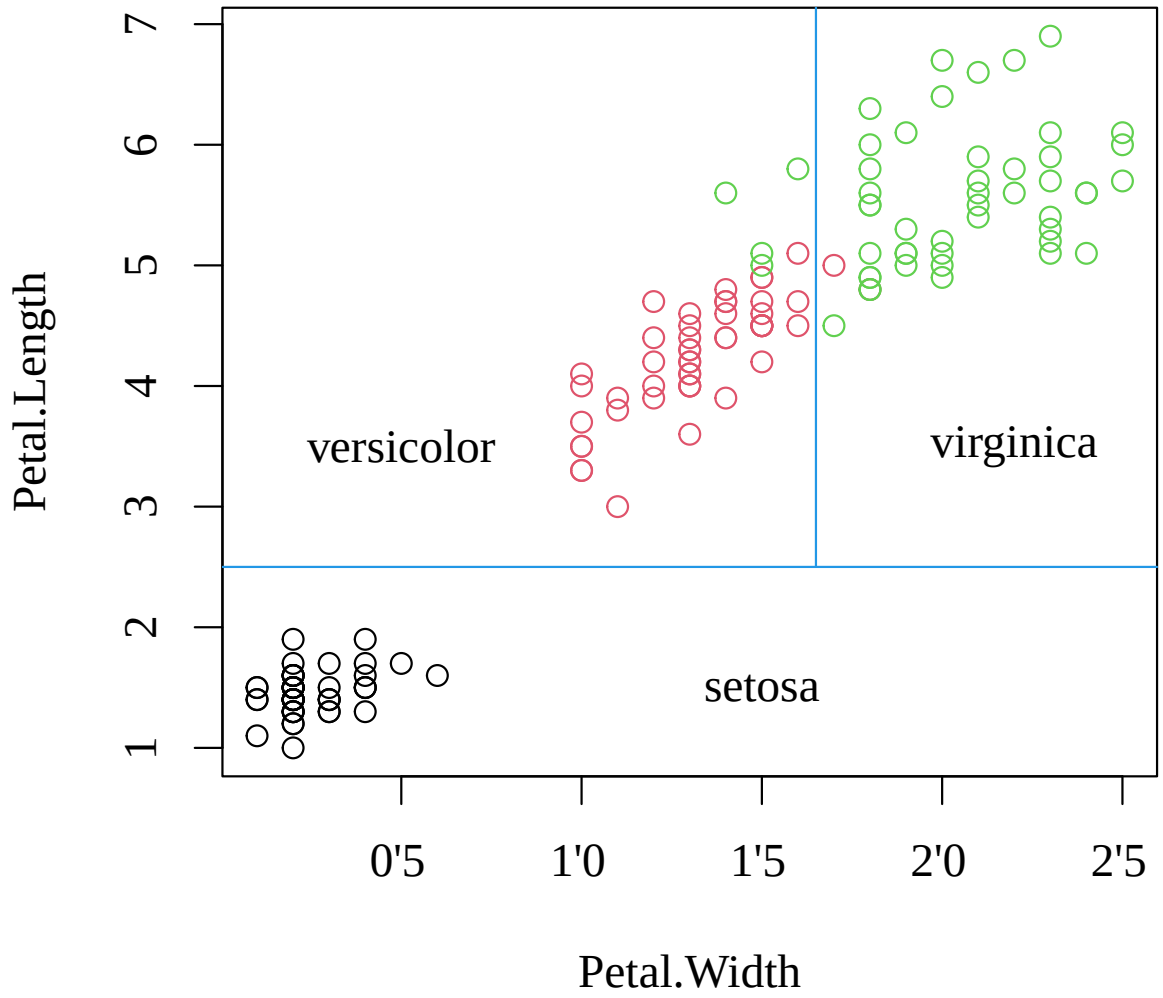

Petal.Length
1
2
3
4
5
6
7
0'5
1'0
1'5
2'0
2'5
Petal.Width
setosa
versicolor
virginica


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

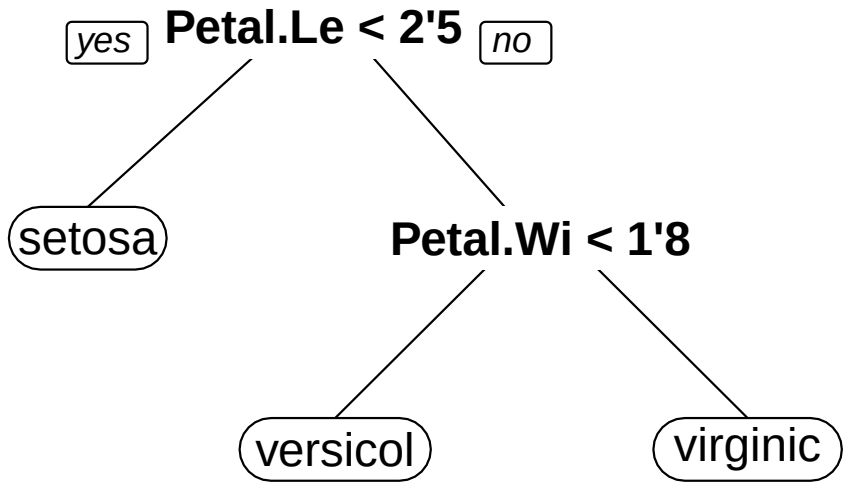

Petal.Length
1
2
3
4
5
6
7
versicolor
virginica
setosa
0'5
1'0
1'5
2'0
2'5
Petal.Width


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

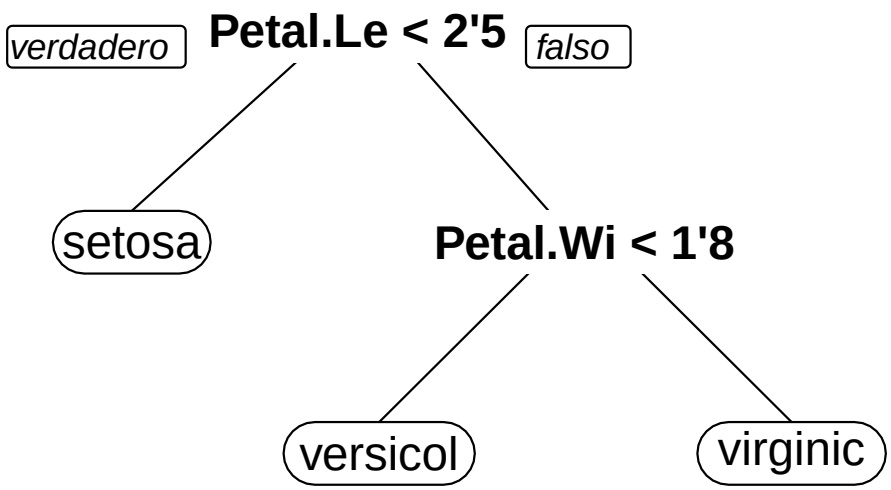
```
> library (rpart.plot) # basta rpart para cálculos
> árbol <- rpart (Species ~
+                   Petal.Length + Petal.Width,
+                 iris)
> árbol
n= 150

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

1) root 150 100 setosa (0'33333 0'33333 0'33333)
  2) Petal.Length< 2.45 50   0 setosa (1'00000 0'00000 0'00000
  3) Petal.Length>=2.45 100  50 versicolor (0'00000 0'50000 0'
    6) Petal.Width< 1.75 54   5 versicolor (0'00000 0'90741 0'
    7) Petal.Width>=1.75 46   1 virginica (0'00000 0'02174 0'9
```


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

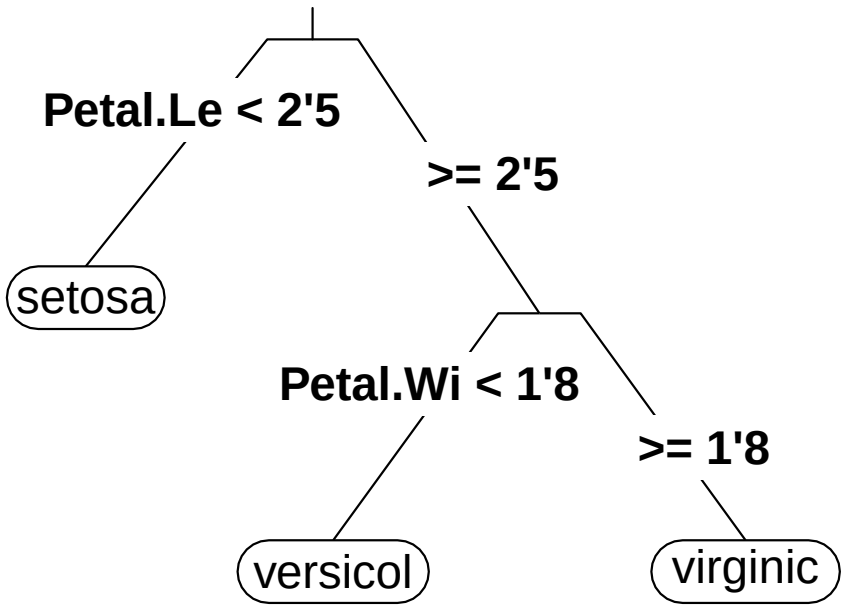
```
> prp (árbol)
```



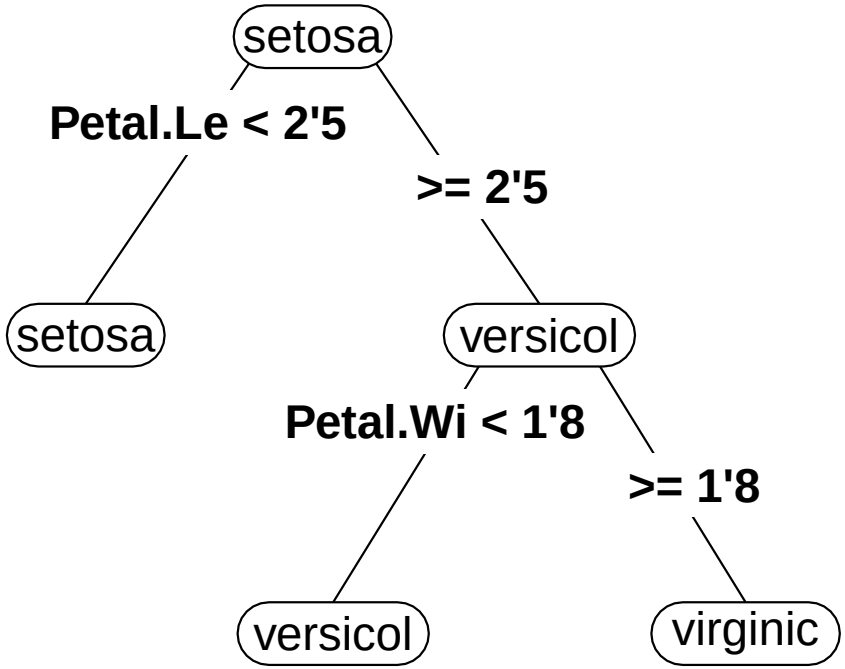
```
> prp (árbol, yes.text="verdadero", no.text="falso")
```



```
> 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*
  - los nodos finales se llaman *hojas*; representan una partición de la muestra
- nodo t: regla de decisión asociada a cierta X_i
  - X_i cuantitativa: ¿ $X_i \leq c_t$?
  - X_i cualitativa: ¿ $X_i \in A_{it} \subset \{a_1, ..., a_m\} = \{\text{valores de } X_i\}$?

Árbol con predictor cualitativo

```
> library (carData)
> rpart (vote ~ region + age, Chile)
n=2532 (168 observations deleted due to missingness)

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

 1) root 2532 1643 N (0'07385 0'35111 0'23223 0'34281)
   2) age< 24.5 564  297 N (0'09220 0'47340 0'14894 0'28546) *
   3) age>=24.5 1968 1261 Y (0'06860 0'31606 0'25610 0'35925)
     6) region=C,SA 1133  740 N (0'06620 0'34687 0'28155 0'305
      12) age< 38.5 453  271 N (0'08609 0'40177 0'28035 0'2317
      13) age>=38.5 680  439 Y (0'05294 0'31029 0'28235 0'3544
     7) region=M,N,S 835  474 Y (0'07186 0'27425 0'22156 0'432
```

Algoritmos

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

CART 1984. Classification And Regression Trees. Ramificaciones binarias. Base de bosques aleatorios.

ID3→C4.5 1986-93. El más popular en Weka (llamado J48). Ramificaciones múltiples. Sólo para clasificación. Existe C5.0 pero resultados similares a CART.

MARS 1991. Multivariate Adaptive Regression Splines (lineal a trozos). Sólo regresión. *Alisador* de CART.

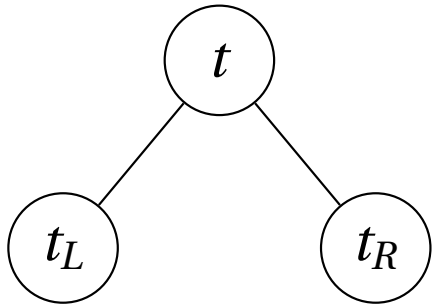
CIT 2006. Conditional Inference Trees. Basado en contrastes de independencia. Soslaya sobreajustes.

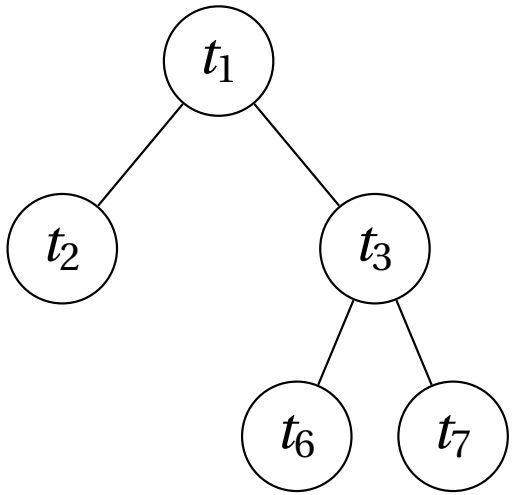
Elementos para la construcción

- método para elegir hoja candidata a división
  - impureza
- criterio de parada
  - riesgo
  - cp (parámetro de complejidad)
  - poda
- método para asignar a un nodo una predicción
  - clasificación: moda
  - regresión: media

Notación de los nodos

- t un cierto nodo
- t_L nodo hijo izquierdo
- t_R nodo hijo derecho

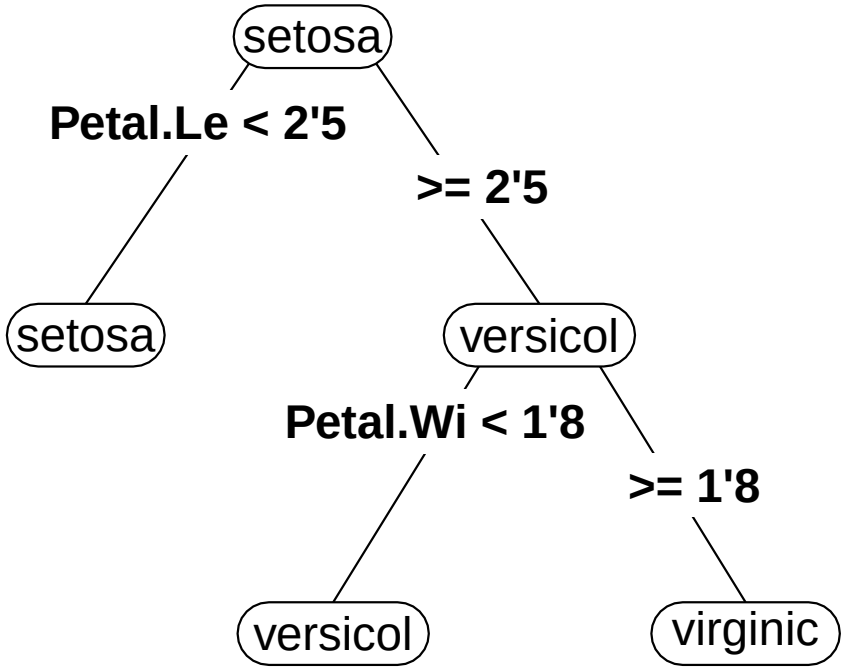




- T conjunto de las hojas del árbol

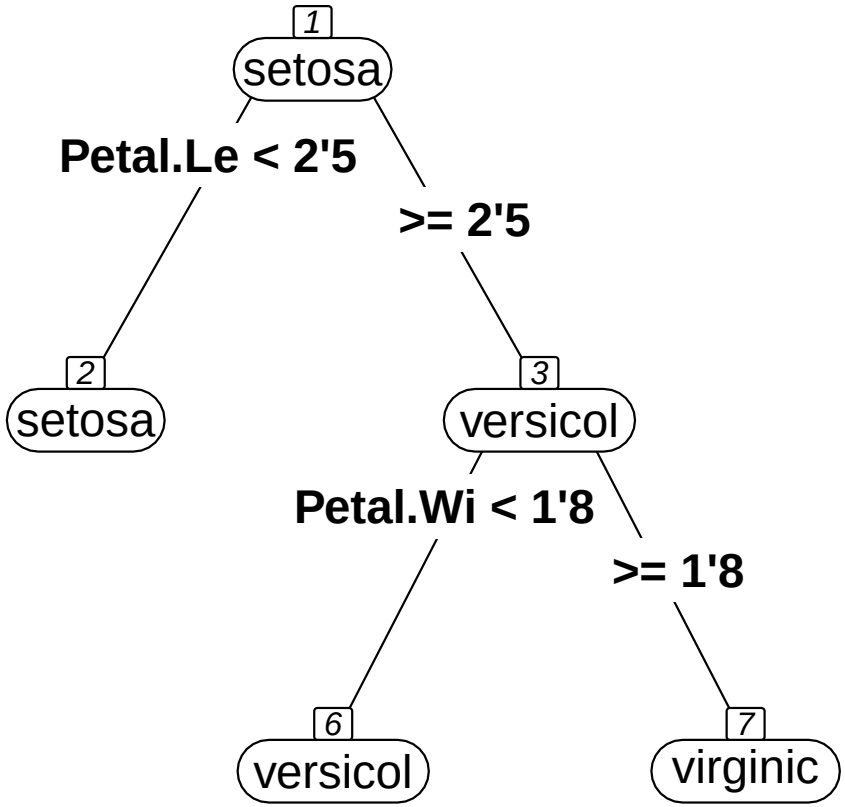
$$T = \{t_2, t_6, t_7\}$$

```
> prp (árbol, type=4)
```

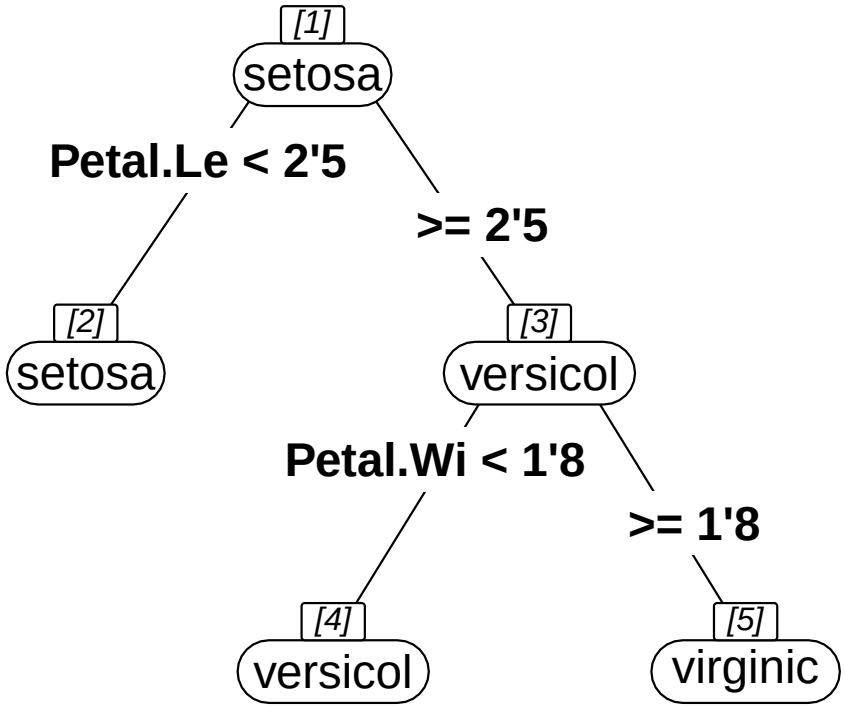


```
> árbol $ frame [, 1:5]          # todos los nodos
          var   n  wt dev yval
1 Petal.Length 150 150 100    1
2       <leaf>  50  50   0    1
3  Petal.Width 100 100  50    2
6       <leaf>  54  54   5    2
7       <leaf>  46  46   1    3
> árbol $ where [c(1:5,145:150)] # hojas
  1   2   3  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

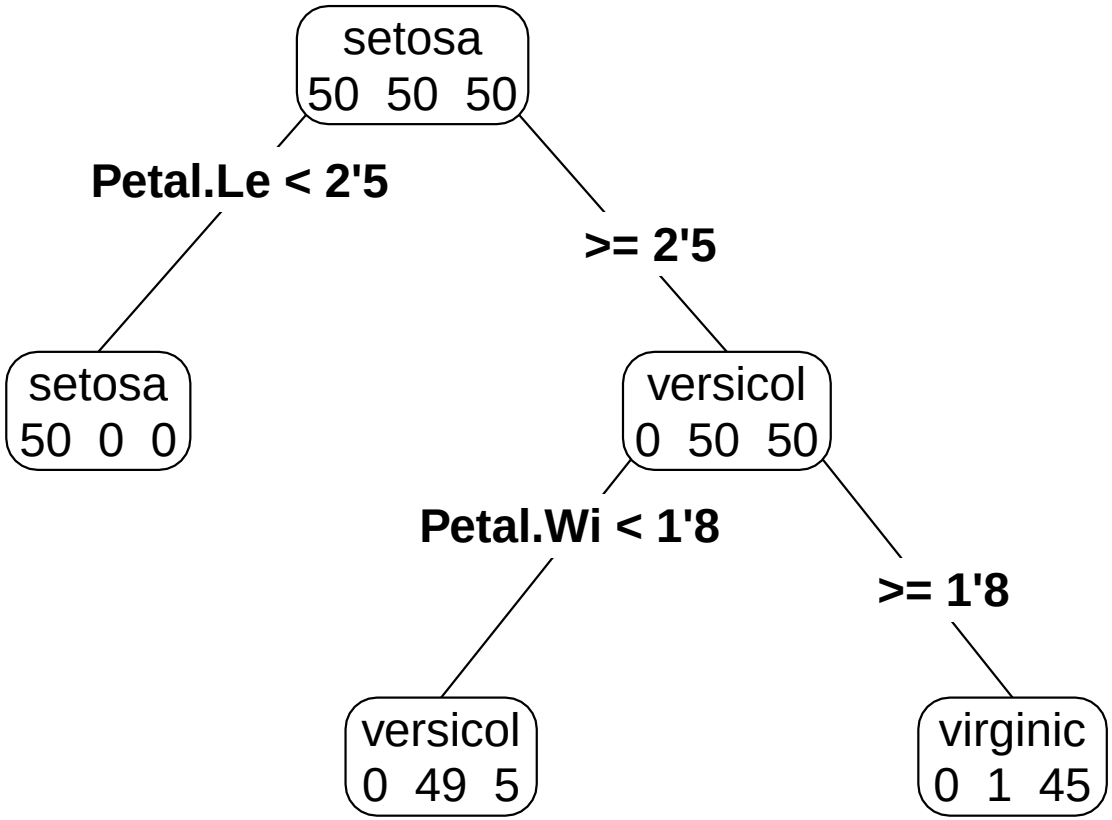
```
> prp (árbol, type=4, nn=TRUE)
```



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



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



Elegir siguiente corte: medidas de impureza

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

donde

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

requisitos:

- alcanzar máximo en $\left(\frac{1}{k}, \ldots, \frac{1}{k}\right)$
- alcanzar mínimo en conjuntos homogéneos como $(0, \ldots, 0, 1, 0, \ldots, 0)$
- invariancia frente a permutaciones de $(p_1, \ldots, p_k)$

- impureza del nodo t

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

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

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

$$\Delta\Phi(t \longrightarrow t_L, t_R) = \Phi(t) - p_L\Phi(t_L) - p_R\Phi(t_R)$$

donde

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

- se busca la regla de decisión (variable X_i y umbral) que maximice $\Delta\Phi$

$$\max_{\substack{i=1,\ldots,p \\ -\infty<c<\infty}} \Delta\Phi(t \xrightarrow{X_i \leq c} t_L, t_R)$$

- impureza del árbol

$$\Phi(T) = \sum_{t \in T} \Pr(t)\Phi(t)$$

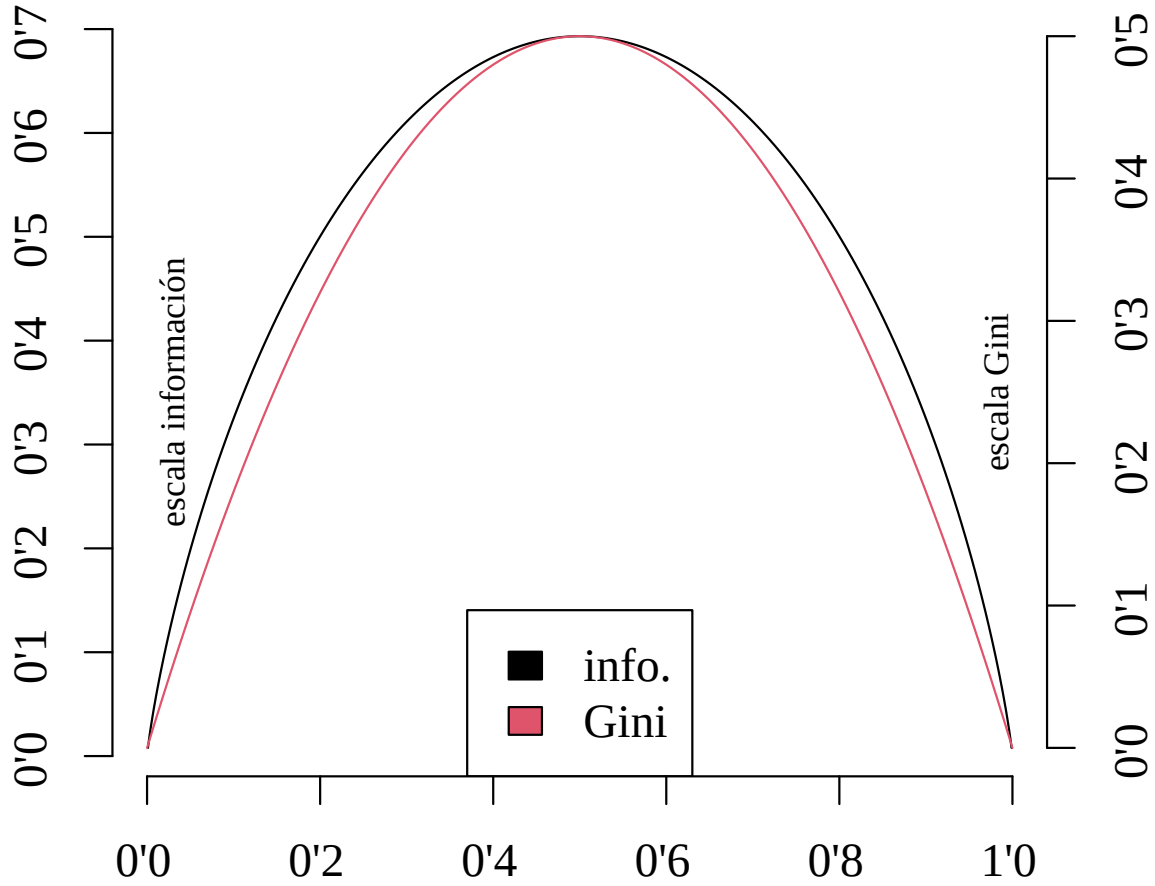
- medidas habituales
  - entropía o información de Shannon

$$\mathrm{H} = -\sum_{j} p_j \log p_j$$

  - Gini

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

Forma de las medidas de impureza (2 clases)



$p_1 = 1 - p_2$

Ejemplo: selección de corte con medidas de impureza

Problema: Predecir si un cliente comprará o no. Datos:

Edad	23	25	30	32	35	38	40	42	45	48
Compra	No	No	Sí	No	Sí	Sí	No	Sí	Sí	Sí

Nodo inicial t: 10 casos (4 `No`, 6 `Sí`)

Probabilidades: $p_{\text{No}} = 0'4$, $p_{\text{Sí}} = 0'6$

$$\text{Entropía: } \mathrm{H}(t) = -0'4 \log_2 0'4 - 0'6 \log_2 0'6 \approx 0'971$$

$$\text{Gini: } G(t) = 1 - (0'4^2 + 0'6^2) = 0'48$$

Ejemplo: selección de corte con medidas de impureza

Posibles cortes (ordenar edad, probar puntos medios):

Corte	t_L (Edad < corte)	t_R (Edad ≥ corte)
27'5	[23,25]: (0 % Sí)	resto: (6/8=75 % Sí)
33'5	[23,25,30,32]: (1/4=25 % Sí)	resto: (5/6=83'3 % Sí)
41'5	hasta 40: (3/7=42'9 % Sí)	desde 42: (3/3=100 % Sí)

	p_L	p_R
27'5	0'2	0'8
33'5	0'4	0'6
41'5	0'7	0'3

	ΔH	ΔG
27'5	$0'971 - 0'2 \cdot 0 - 0'8 \cdot 0'811 = 0'322$	$0'48 - 0'2 \cdot 0 - 0'8 \cdot 0'375 = 0'18$
33'5	$0'971 - 0'4 \cdot 0'811 - 0'6 \cdot 0'650 = 0'081$	$0'48 - 0'4 \cdot 0'375 - 0'6 \cdot 0'278 = 0'097$
41'5	$0'971 - 0'7 \cdot 0'985 - 0'3 \cdot 0 = 0'282$	$0'48 - 0'7 \cdot 0'490 - 0'3 \cdot 0 = 0'137$

Ejemplo: selección de corte con medidas de impureza

conclusión

- el corte en 27'5 maximiza la reducción de entropía ($\Delta \mathrm{H} = 0'322$)
- también maximiza la reducción de Gini ($\Delta G = 0'18$)
- por tanto, se elegiría `Edad < 27.5` como primera división

impureza $\Phi(T)$del árbol final (si no seguimos dividiendo)

- entropía

$$\Pr(t_L)\Phi(t_L) + \Pr(t_R)\Phi(t_R) = 0'2 \cdot 0 + 0'8 \cdot 0'811 = 0'649$$

- Gini

$$0'2 \cdot 0 + 0'8 \cdot 0'375 = 0'3$$

```
> ## 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"))
```

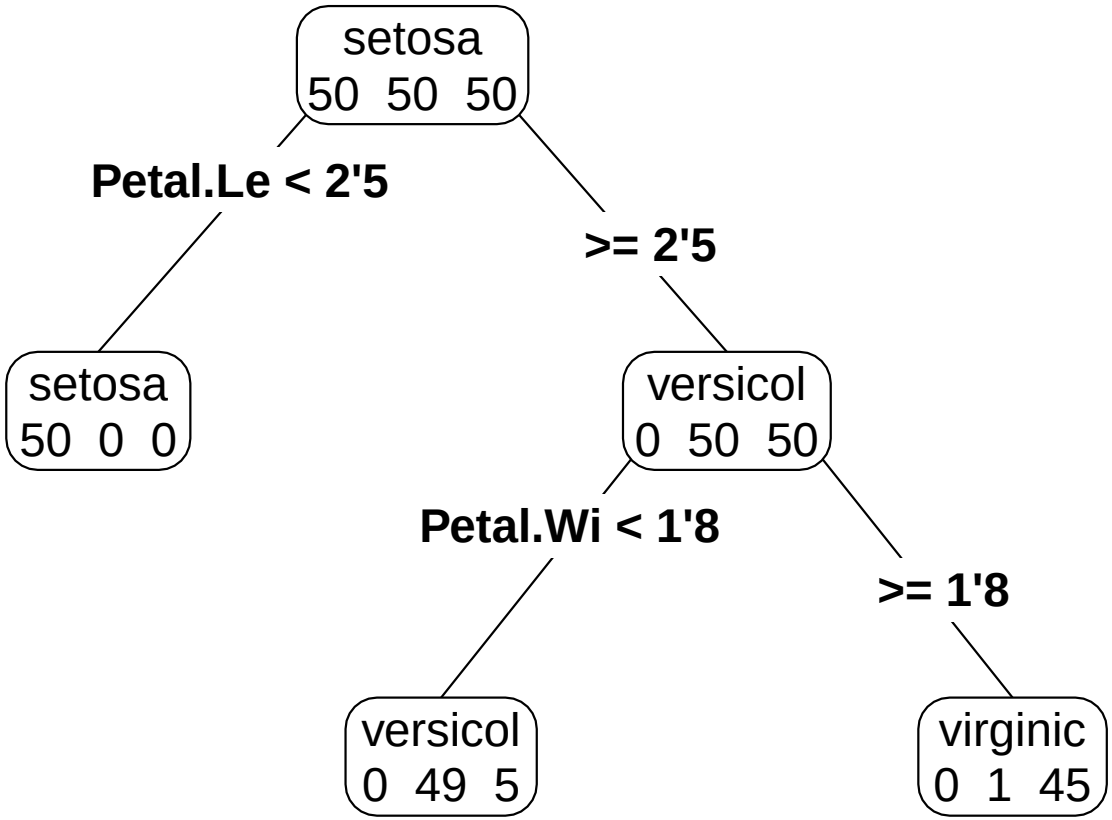
Criterio de parada: Riesgo

- riesgo o pérdida de una hoja

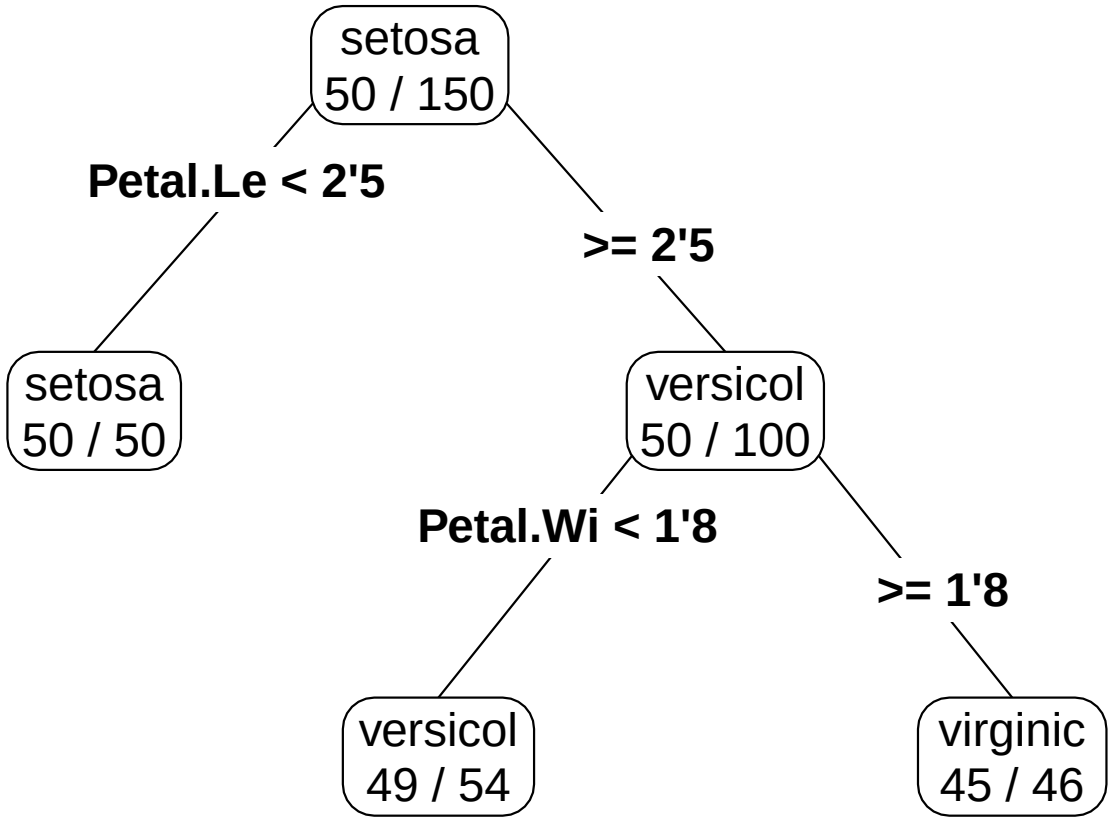
$$\begin{aligned} R(t) &= \Pr[\text{error de clasificación} \mid t] \\ &= 1 - \max_{j \in \text{clases}} \Pr[j \mid t] \end{aligned}$$

- ejemplos: raíces
  - clientes
    - predicción: siempre `Sí` (moda, 6/10)
    - errores: 4 casos `No` $\Rightarrow$ Riesgo $R = \frac{4}{10} = 0'40$
  - iris
    - predicción: siempre `setosa`
      (hay tres modas, 50/150; escogida por orden alfabético)
    - errores: 50 `versicolor` y 50 `virginica` $\Rightarrow R = 100/150 = 0'67$

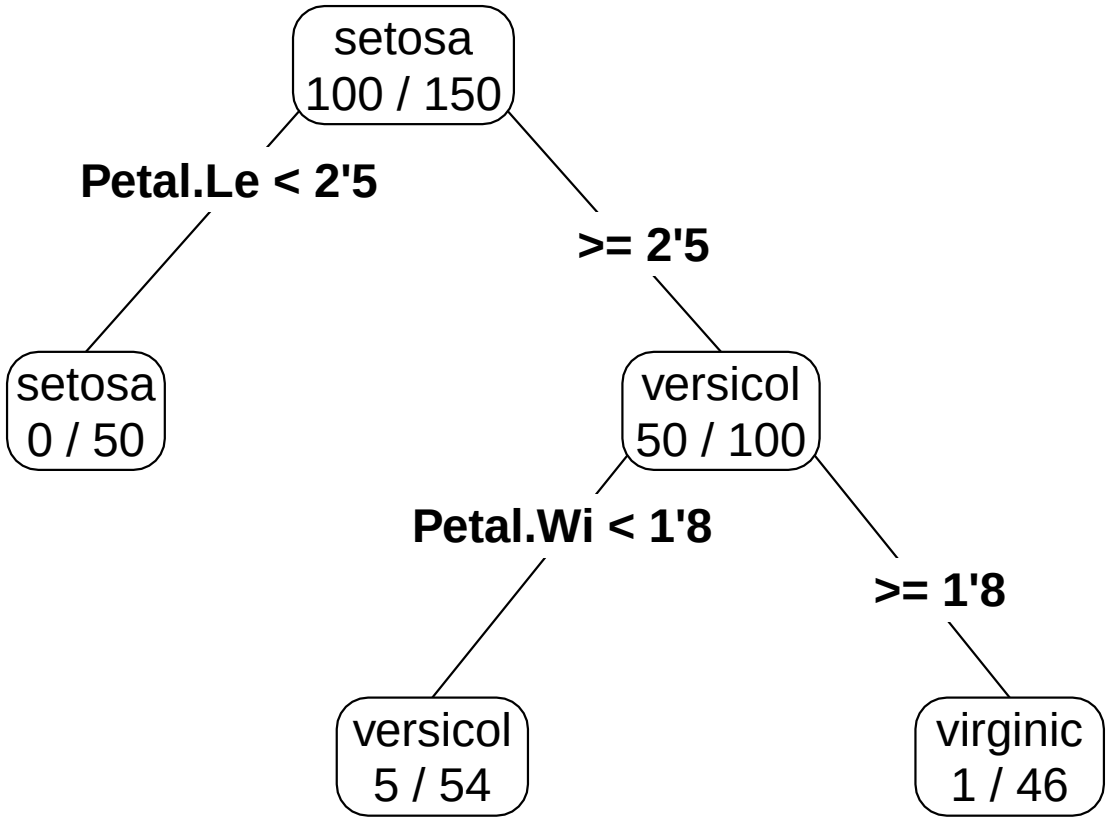
```
> prp (árbol, type=4, extra=1)
```



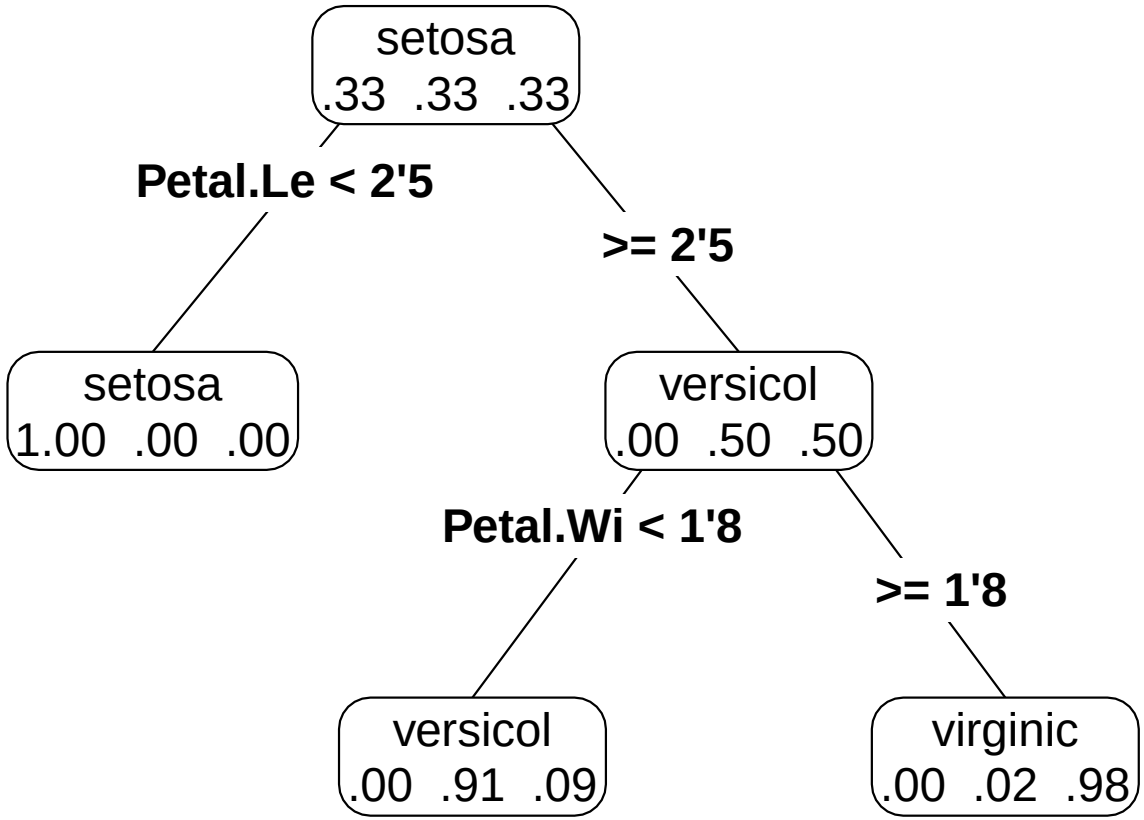
```
> prp (árbol, type=4, extra=2)
```



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



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



Criterio de parada: 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

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

- ejemplo: T = (árbol de clientes con un corte en Edad $< 27'5$)
  - nodo izquierdo t_L (edad $< 27'5$): casos {23,25}, ambos `No` ; predicción `No`, errores = 0
  - nodo derecho t_R (edad $\geq 27'5$): 8 casos, 6 `Sí` y 2 `No` ; predicción `Sí`, errores = 2

$$R(T) = \frac{|t_L|}{N} \cdot R(t_L) + \frac{|t_R|}{N} \cdot R(t_R) = \frac{2}{10} \cdot 0 + \frac{8}{10} \cdot \frac{2}{8} = 0'20$$

Criterio de parada: 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 $t \longrightarrow \{t_L, t_R\}$

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

- ¿ criterio de corte: maximizar ΔR ?

¿Riesgo como criterio de corte?

- ¿ maximizar ΔR ?
- ejemplo 1:
  - supóngase 80 % de individuos de clase 1 en la raíz
  - supóngase corte candidato con hijos equiprobables
    - 60 % 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
- ejemplo 2: sean dos cortes candidatos con hojas equiprob.
  - 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 ; $R(A) = 0'3 < R(B) = 0'325$
  - el B es preferible en la práctica,
    porque establece mejor cómo seguir dividiendo
  - mediante impurezas, se prefiere B
    - Gini: $G(A) = 0'42$; $G(B) = 0'3775$
    - información: $\mathrm{H}(A) = 0'881$; $\mathrm{H}(B) = 0'805$

Criterios de parada en `rpart` de R

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

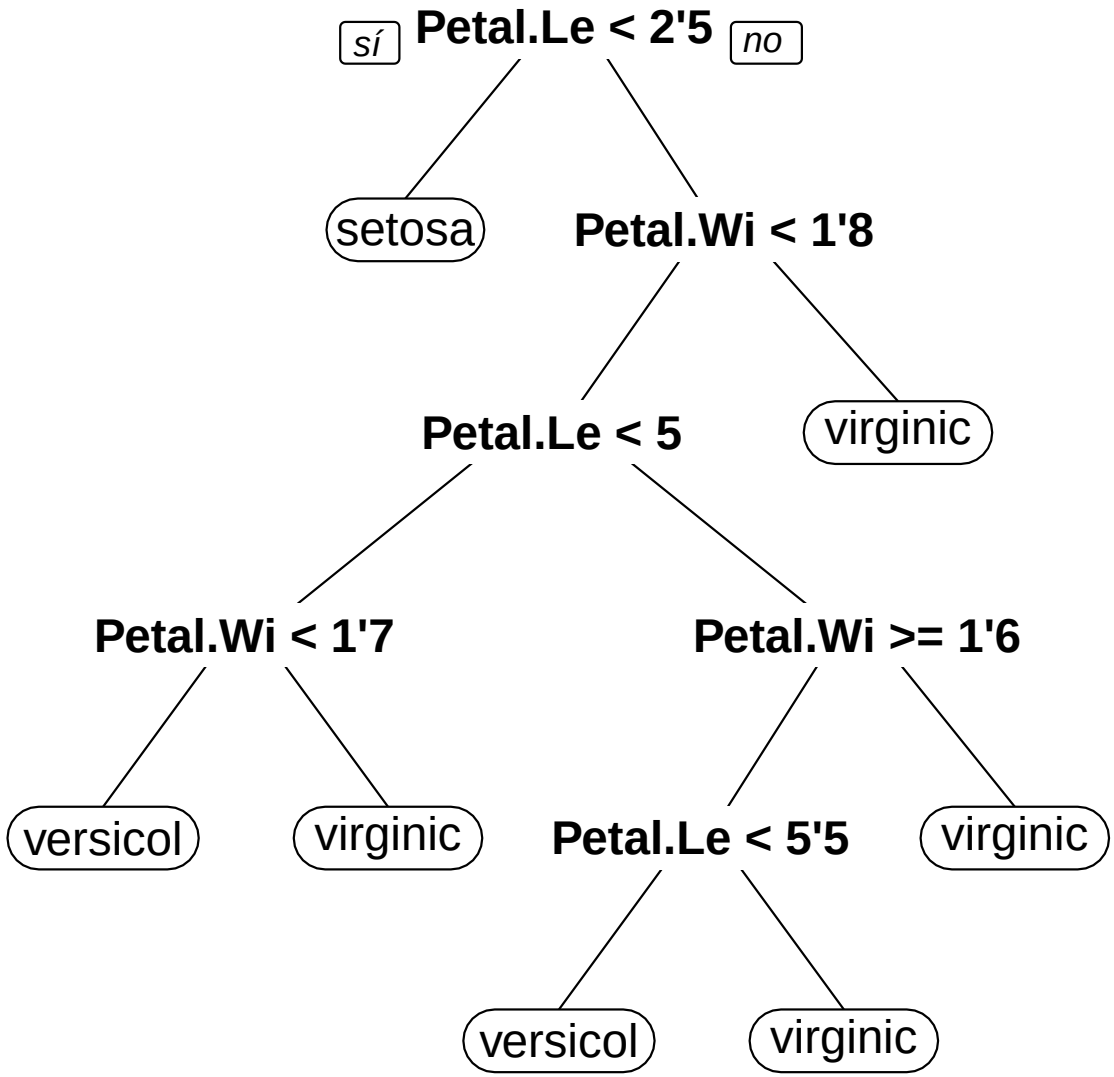
minbucket Tamaño mínimo de cada hoja. Por omisión, 7.

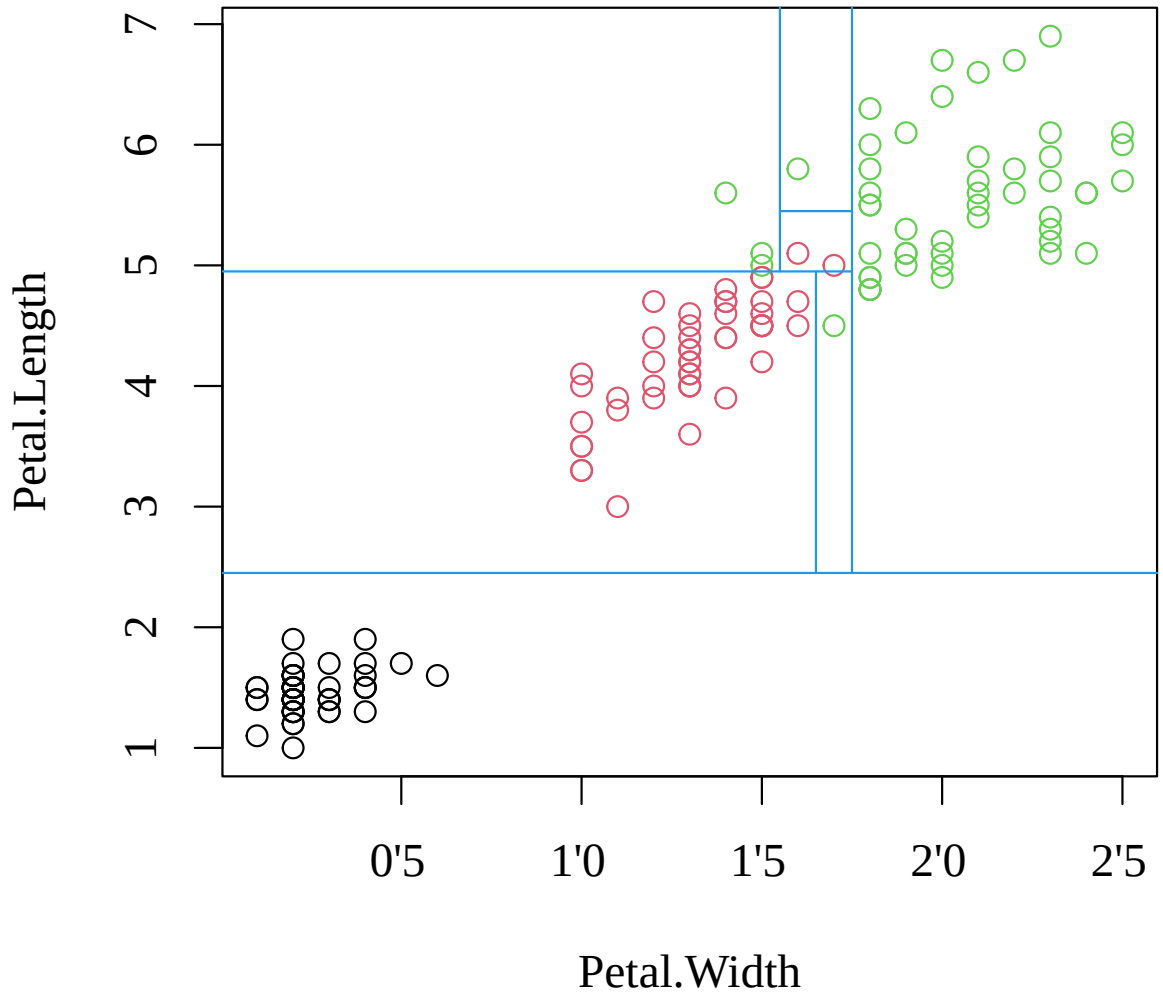
cp Parámetro de complejidad.
Coeficiente de penalización por número de hojas.
Por omisión, 1 %.

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 (0'33333 0'33333 0'33333)
   2) Petal.Length< 2.45 50   0 setosa (1'00000 0'00000 0'0000
   3) Petal.Length>=2.45 100  50 versicolor (0'00000 0'50000 0
     6) Petal.Width< 1.75 54   5 versicolor (0'00000 0'90741 0
      12) Petal.Length< 4.95 48   1 versicolor (0'00000 0'9791
        24) Petal.Width< 1.65 47   0 versicolor (0'00000 1'000
        25) Petal.Width>=1.65 1   0 virginica (0'00000 0'00000
      13) Petal.Length>=4.95 6   2 virginica (0'00000 0'33333
        26) Petal.Width>=1.55 3   1 versicolor (0'00000 0'6666
          52) Petal.Length< 5.45 2   0 versicolor (0'00000 1'0
          53) Petal.Length>=5.45 1   0 virginica (0'00000 0'00
        27) Petal.Width< 1.55 3   0 virginica (0'00000 0'00000
     7) Petal.Width>=1.75 46   1 virginica (0'00000 0'02174 0'
```




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

Criterio de parada cp

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

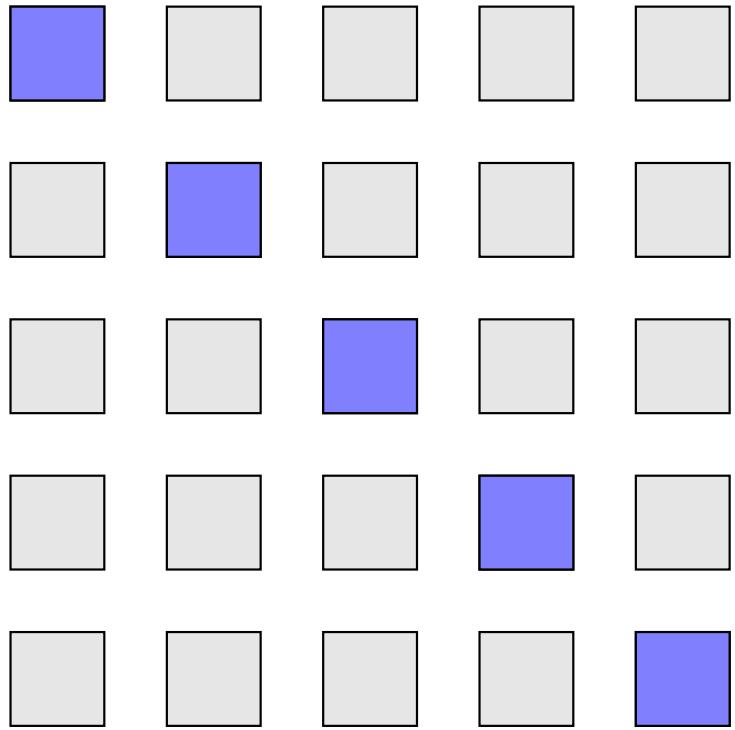
```
> printcp (árbol)                    # árbol$cptable
Classification tree:
rpart(formula = Species ~ Petal.Length + Petal.Width, data = i

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

Root node error: 100/150 = 0'67

n= 150

    CP nsplit rel error xerror  xstd
1 0'50      0      1'00   1'16 0'051
2 0'44      1      0'50   0'68 0'061
3 0'01      2      0'06   0'09 0'029
```

Validación cruzada

- objetivo: estimar el error en datos nuevos
- dividir la muestra en K partes (pliegues)
- para cada $k = 1, \ldots, K$:
  - entrenar con $K - 1$ partes
  - evaluar en la parte restante

$$\text{Error}_{\text{VC}} = \frac{1}{K} \sum_{k=1}^{K} \text{Error}_k$$

- En `rpart`: argumento `xval`



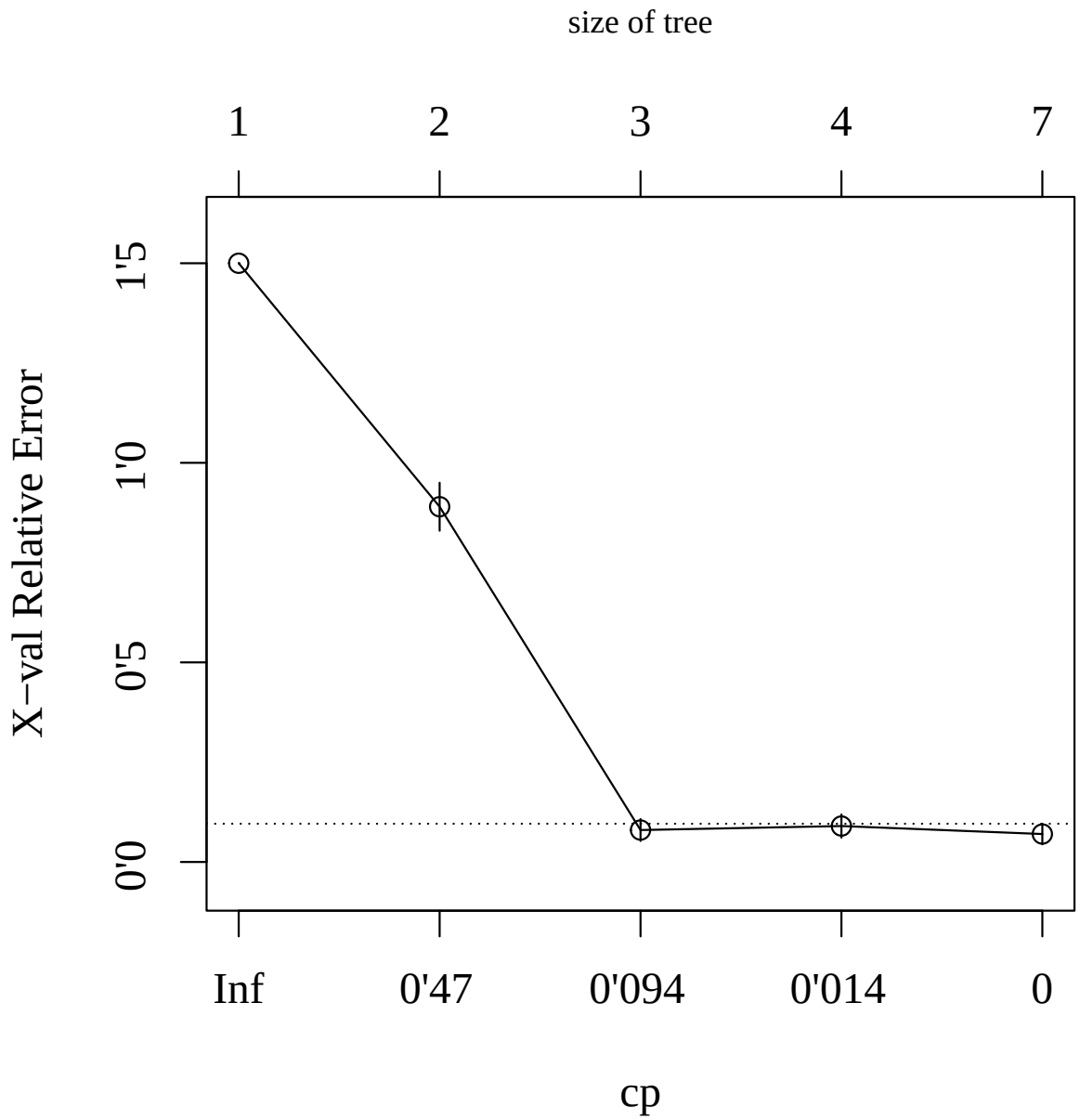
cada fila: un pliegue usado como validación (azul) y cuatro como entrenamiento (gris)

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 0'00000
2 0'44      1      0'50   0'89 0'06016
3 0'02      2      0'06   0'08 0'02752
4 0'01      3      0'04   0'09 0'02909
5 0'00      6      0'01   0'07 0'02583
```

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

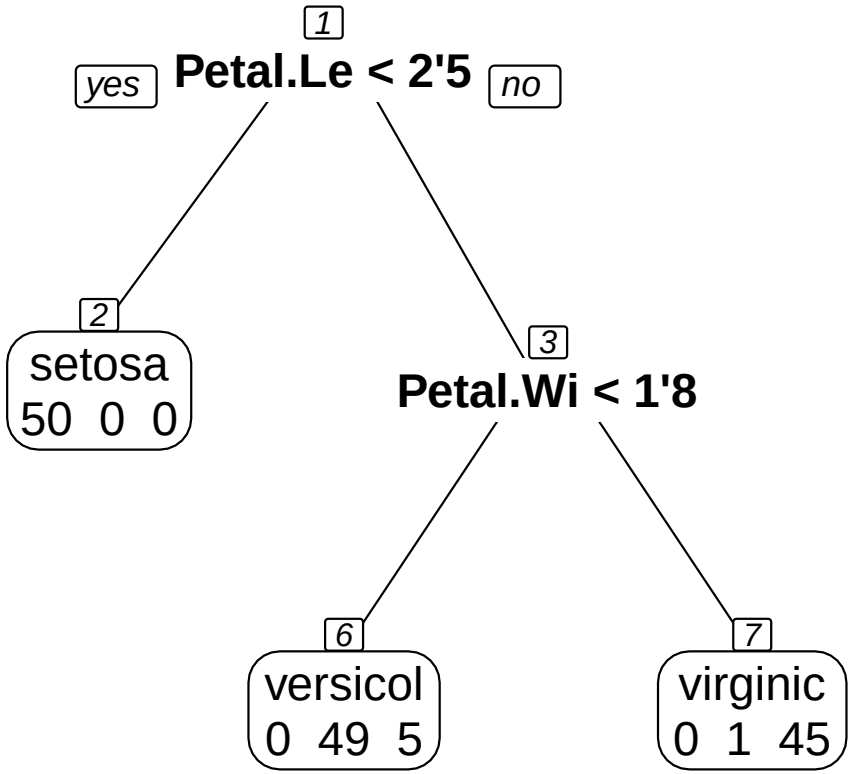


Poda

- si árbol demasiado ralo,
  - obtener T_0 con `cp=0, minbucket=1`
  - chequear `printcp` o `plotcp` y podar según `cp` óptimo

```
> 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 (0'33333 0'33333 0'33333)
  2) Petal.Length< 2.45 50   0 setosa (1'00000 0'00000 0'00000
  3) Petal.Length>=2.45 100  50 versicolor (0'00000 0'50000 0'
    6) Petal.Width< 1.75 54   5 versicolor (0'00000 0'90741 0'
    7) Petal.Width>=1.75 46   1 virginica (0'00000 0'02174 0'9
```

Importancia de las variables



Importancia de las variables

- suma de calidad de la división en nodos cuyas reglas involucran la variable X_i : $\sum_{t \text{ usa } X_i} \Delta\Phi(t)$
- si dos variables están muy correladas, su importancia conjunta se dividirá entre las dos y puede que ninguna destaque
- como componente `variable.importance` del objeto `rpart` aparece en forma absoluta

```
> árbol $ variable.importance
 Petal.Width Petal.Length
       88'97        81'34
```

- en la salida de `summary` aparece en porcentajes

Importancia de las variables

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

    CP nsplit rel error xerror    xstd
1 0'50      0      1'00   1'16 0'05128
2 0'44      1      0'50   0'68 0'06097
3 0'01      2      0'06   0'09 0'02909

Variable importance
 Petal.Width Petal.Length
          52           48

Node number 1: 150 observations,    complexity param=0'5
  predicted class=setosa      expected loss=0.6667  P(node) =1
    class counts:    50    50    50
```

Regresión

- Y cuantitativa
- `rpart` usa el método `anova`
- 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 moda)
- 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
- 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` = ¿10, 100, . . . ?
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 (árbol)          # o printcp(árbol)
```

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

```
árbol1 <- prune (árbol, cp=...)
árbol1 <- 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 validación cruzada.

Ejercicio 3: solder

- Construye un árbol a partir de los datos `solder.balance` 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`.

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

Apéndice

- muestras desbalanceadas o desequilibradas
- p.ej. prevalencia baja

Impureza de Gini (caso estándar)

Sea un nodo con probabilidades:

$$p_k = P(Y = k \mid \text{nodo})$$

Índice de Gini:

$$\Phi = \sum_k p_k(1 - p_k) = 1 - \sum_k p_k^2$$

Interpretación:

- Probabilidad de error al clasificar aleatoriamente según p_k
- Equivalente a pérdida 0-1 simétrica (aproximación suave)

Riesgo bayesiano con pérdidas

Sea una matriz de pérdidas:

$$L(i, j)$$

Riesgo al predecir j:

$$R(j) = \sum_k L(k, j)\, p_k$$

Riesgo óptimo en el nodo:

$$\Phi = R^* = \min_j R(j)$$

Interpretación:

- Generaliza la impureza
- Incorpora asimetría de costes

Gini como caso particular

Si:

$$L(i,j) = \mathbf{1}_{i \neq j}$$

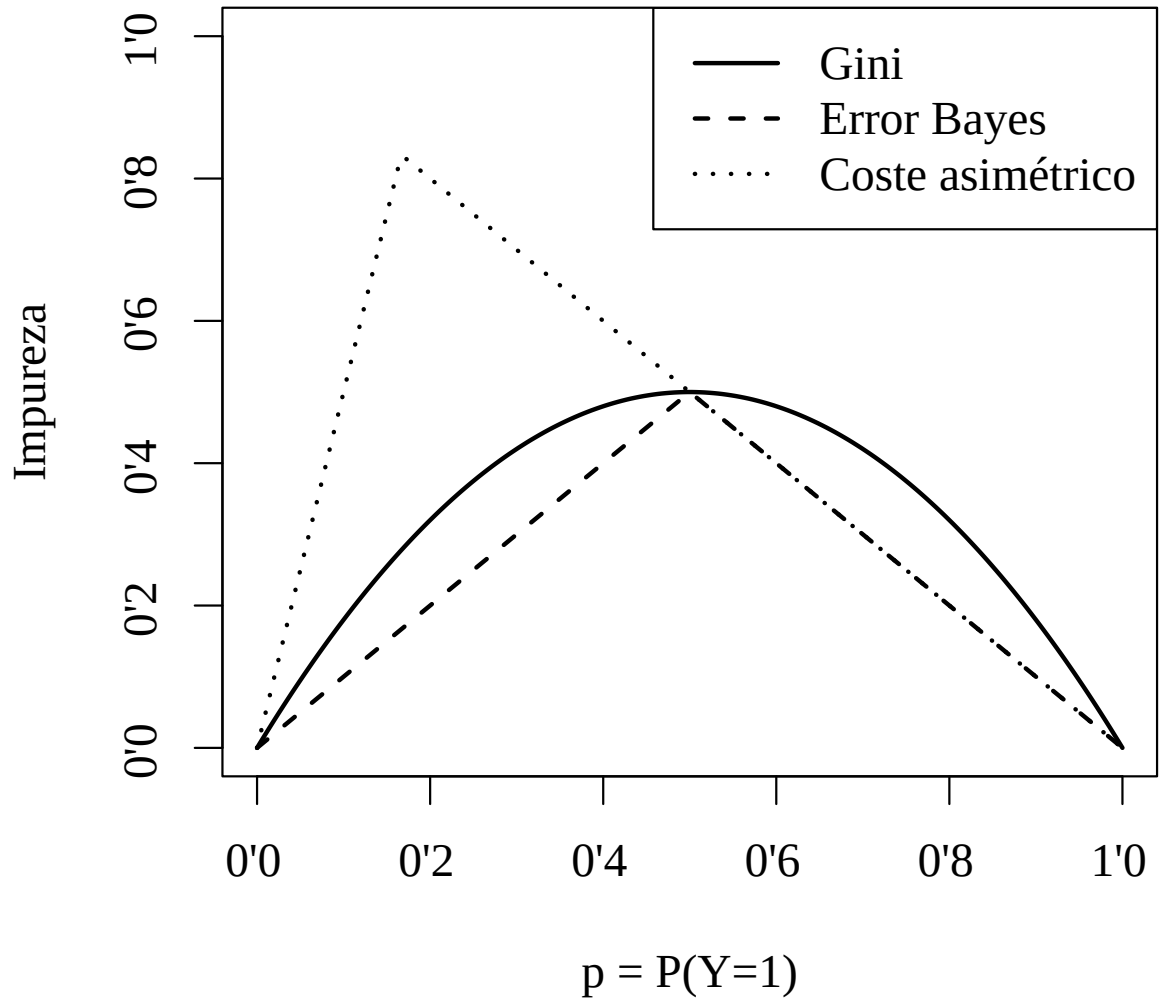
entonces:

$$R(j) = 1 - p_j$$

$$\Phi = R^* = 1 - \max_k p_k$$

Comparación:

- $\Phi = 1 - \max_k p_k$: error Bayes óptimo
- $\Phi = 1 - \sum_k p_k^2$: Gini (versión suavizada)



Impureza generalizada (con pérdidas)

$$\Phi = \min_j \sum_k L(k,j)\, p_k$$

Idea clave:

- La impureza es el riesgo mínimo
- Depende de la decisión óptima
- Generaliza Gini y error de clasificación

Caso binario explícito

Clases: $\{0, 1\}$

$$L = \begin{pmatrix} 0 & c_{\text{FP}} \\ c_{\text{FN}} & 0 \end{pmatrix}$$

Sea $p = P(Y = 1)$:

$$R(0) = c_{\text{FN}}\, p \quad R(1) = c_{\text{FP}}\, (1 - p)$$

$$\Phi(p) = \text{mín}\{c_{\text{FN}} p,\ c_{\text{FP}}(1 - p)\}$$

Forma de la impureza

$$\Phi(p) = \text{mín}\{c_{\text{FN}}\, p,\ c_{\text{FP}}(1-p)\}$$

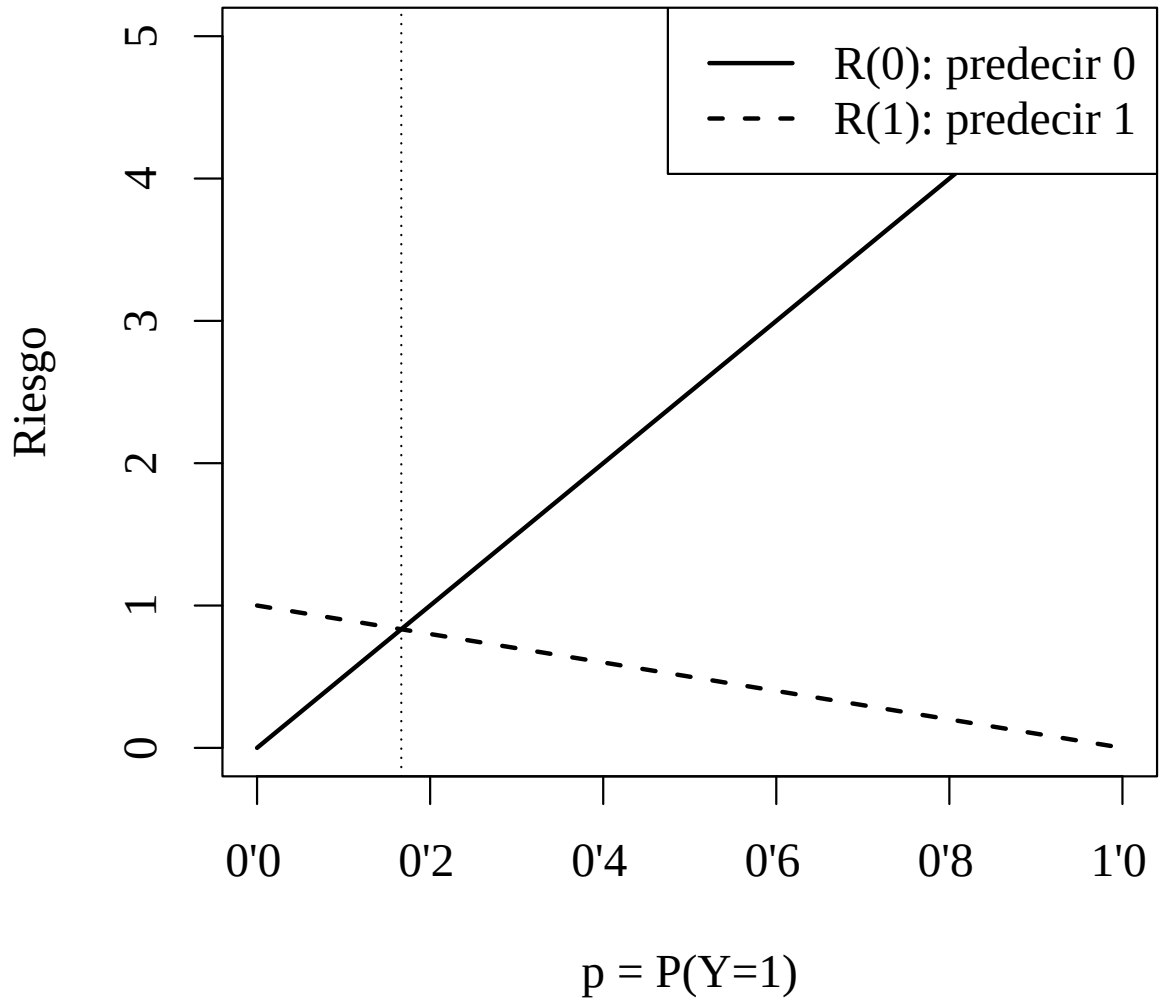
Propiedades:

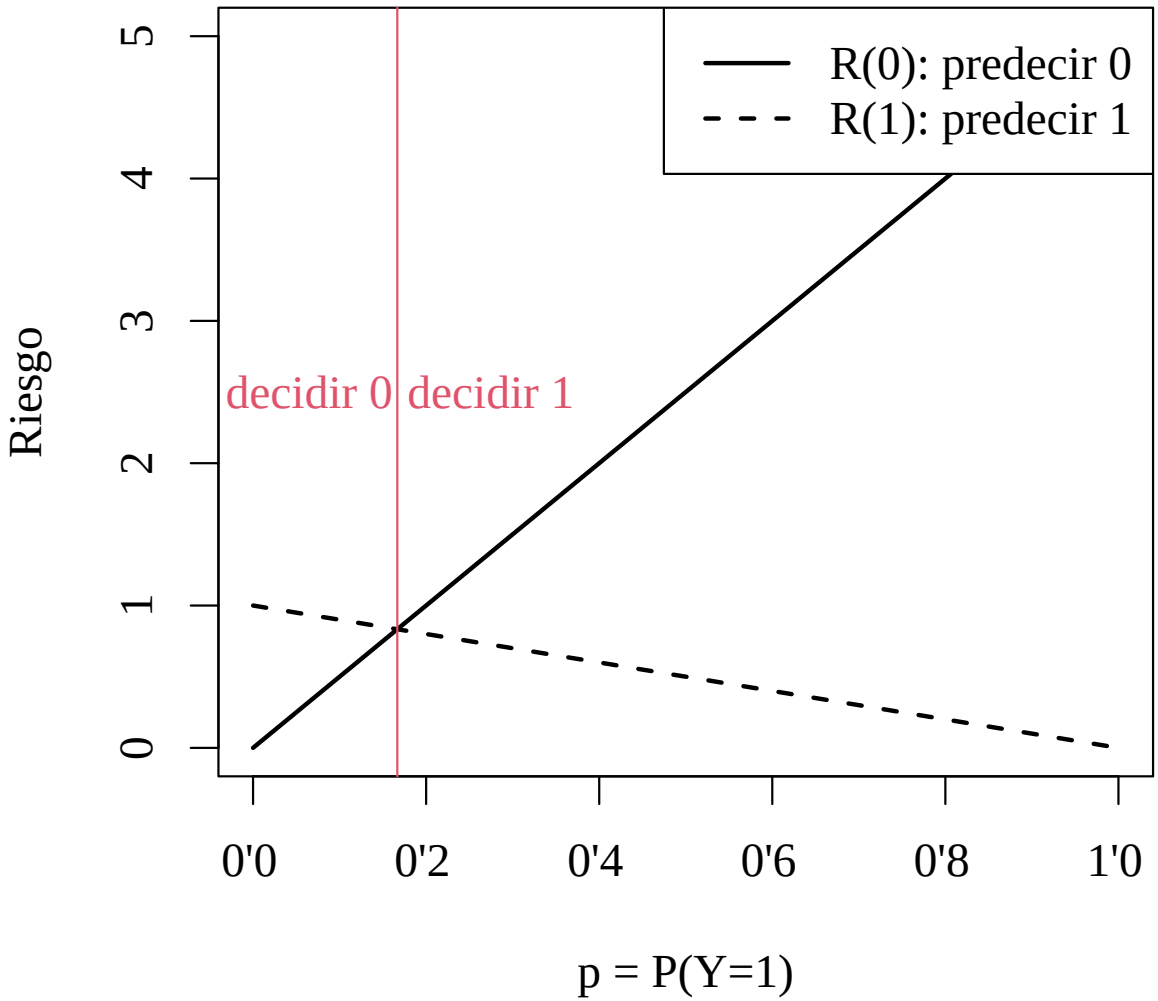
- Función cóncava a trozos
- No simétrica salvo $c_{\text{FN}} = c_{\text{FP}}$
- Máximo en:

$$p = \frac{c_{\text{FP}}}{c_{\text{FP}} + c_{\text{FN}}}$$

Comparación:

- Gini estándar: máximo en $p = 1/2$
- Φ: máximo desplazado


R(0): predecir 0
R(1): predecir 1
Riesgo
0
1
2
3
4
5
0'0
0'2
0'4
0'6
0'8
1'0
p = P(Y=1)


R(0): predecir 0
R(1): predecir 1
decidir 0
decidir 1
Riesgo
0
1
2
3
4
5
0'0
0'2
0'4
0'6
0'8
1'0
p = P(Y=1)

Incorporación de `prior`

Sea π_k el prior:

$$\tilde{p}_k \propto \pi_k \, \hat{p}_k$$

Impureza:

$$\Phi = \min_j \sum_k L(k,j) \, \pi_k \, \hat{p}_k$$

Interpretación:

- π_k reescala las frecuencias
- Cambia la distribución efectiva

Criterio de división

Para un split:

$$\Delta = \Phi(\text{padre}) - \left(\frac{n_L}{n} \Phi(L) + \frac{n_R}{n} \Phi(R) \right)$$

Optimización:

- Maximizar reducción de impureza
- Con impureza definida como riesgo

Conclusión:

`rpart` implementa CART basado en riesgo bayesiano

Problema: `kyphosis`

Contexto:

- Pacientes pediátricos (edad ≤ 200 meses)
- Variable respuesta: `Kyphosis` $\in \{$`absent`, `present`$\}$
- Objetivo: predecir recurrencia tras cirugía

Distribución de clases:

$$n = 81, \quad n_{\texttt{absent}} = 64, \quad n_{\texttt{present}} = 17$$

Desbalanceo:

$$P(\texttt{absent}) \approx 0'79, \quad P(\texttt{present}) \approx 0'21$$

Clasificador trivial

Regla: predecir siempre absent

$$\hat{Y} = \texttt{absent} \quad \forall x$$

Error:

$$\text{error} = P(Y = \texttt{present}) \approx 0'21$$

$$\text{accuracy} \approx 79\,\%$$

Conclusión:

- Alta accuracy sin usar variables
- No detecta ningún caso positivo

Limitaciones del enfoque estándar

Problema:

- La accuracy no refleja utilidad clínica

Falso negativo:

- No detectar cifosis
- Posible impacto clínico grave

Falso positivo:

- Intervención innecesaria
- Coste menor (relativo)

$$c_{\text{FN}} \gg c_{\text{FP}}$$

Motivación para `prior` y `loss`

Objetivo: optimizar riesgo, no accuracy

$$\Phi = \min_j \sum_k L(k,j)\, \pi_k\, \hat{p}_k$$

Estrategias:

- Ajustar π_k (prior) para corregir desbalanceo
- Ajustar $L(k,j)$ para reflejar costes

Resultado esperado:

- Mayor sensibilidad (detectar `present`)
- Menor dependencia de la mayoría

Clasificador trivial: matriz de confusión

Regla: predecir siempre absent

$$\hat{Y} = \texttt{absent} \quad \forall x$$

Matriz de confusión:

	Pred: absent	Pred: present
Real: absent	64	0
Real: present	17	0

Métricas:

$$\text{accuracy} = \frac{64}{81} \approx 0'79 \qquad \text{sensibilidad} = 0$$

Comparación conceptual: accuracy vs riesgo

Clasificador trivial:

- Alta accuracy ($\approx 79\,\%$)
- Sensibilidad nula

Problema:

Accuracy no incorpora costes asimétricos

Enfoque con pérdidas:

$$\Phi = \min_j \sum_k L(k,j)\, \pi_k\, \hat{p}_k$$

Efecto:

- Penaliza fuertemente falsos negativos
- Fuerza modelos a detectar `present`

Ejemplo base en R

Dataset: `kyphosis`

```
library(rpart)

fit0 <- rpart(Kyphosis ~ ., data = kyphosis,
              method = "class")
```

Configuración implícita:

- Prior empírico
- Pérdida 0-1 simétrica

$$\Phi = 1 - \max_k p_k \quad \text{(aprox. Gini)}$$

Uso de prior uniforme

Objetivo: compensar desbalanceo

```
fit1 <- rpart(Kyphosis ~ ., data = kyphosis,
              method = "class",
              parms = list(prior = c(0.5, 0.5)))
```

Efecto:

- Repondera clases:

$$\tilde{p}_k \propto \pi_k \hat{p}_k$$

- Mayor atención a `present`

Consecuencia:

- Splits más sensibles a la clase minoritaria

Matriz de pérdidas asimétrica

Objetivo: penalizar falsos negativos

$$L = \begin{pmatrix} 0 & 1 \\ 5 & 0 \end{pmatrix}$$

```
L <- matrix(c(0,1,
              5,0), nrow = 2, byrow = TRUE)

fit2 <- rpart(Kyphosis ~ ., data = kyphosis,
              method = "class",
              parms = list(loss = L))
```

Efecto:

$$\Phi(p) = \text{mín}\{5p,\ (1-p)\}$$

Combinación: prior + loss

Modelo más realista

```
L <- matrix(c(0,1,
              5,0), 2, byrow = TRUE)

fit3 <- rpart(Kyphosis ~ ., data = kyphosis,
              method = "class",
              parms = list(
                prior = c(0.5, 0.5),
                loss  = L))
```

Impureza:

$$\Phi = \min_j \sum_k L(k,j)\, \pi_k\, \hat{p}_k$$

Interpretación:

- Prior: frecuencia esperada
- Loss: coste decisional

Comparación de modelos

```
rpart.plot::rpart.plot(fit0)
rpart.plot::rpart.plot(fit2)
```

Observar:

- Cambios en los splits
- Cambios en las hojas (clase predicha)
- Mayor sensibilidad en `fit2`, `fit3`

Mensaje clave:

El árbol depende de Φ, no solo de los datos