

# Análisis de Conglomerados



José A. Perusquía Cortés  
Análisis Multivariado, Semestre 2025-II



# Motivación

- ▶ Clustering es el procesos de **agrupar objetos similares** buscando patrones en los datos
- ▶ Técnica de aprendizaje no supervisado, i.e., a priori no necesitamos:
  - Conocer el número de clusters (en algunas ocasiones se puede conocer)
  - Un grupo de observaciones etiquetadas (training data)
- ▶ Dos tipos de métodos
  - Si no se conoce el número de clusters se tienen métodos jerárquicos aglomerativos y divisivos
  - Si se conoce el número de clusters se crean particiones (cada objeto pertenece a un cluster) o métodos “fuzzy” (cada objeto puede pertenecer a varios clusters)

# Clusters

- Necesitamos definir una noción de **cercanía**
  - Matriz de distancias
  - Matriz de disimilitudes
  - Matriz de similitudes
  
- En la práctica es común utilizar
  - Distancia euclidiana
  - Distancia Manhattan
  - Distancia Mahalanobis

# Métodos jerárquicos aglomerativos (AGNES)

# Métodos aglomerativos

- ▶ **¿Qué necesitamos?**

- Una matriz de proximidades (e.g. distancias, disimilitudes)
- Medida de distancia entre clusters

- ▶ **Idea**

Crear un árbol de clusters empezando con  $n$  grupos de una sola observación y unirlos uniéndolos por cercanía

- ▶ **¿Cómo medir la distancia entre clusters?**

# Vecino más cercano

- ▶ También conocido como **single linkage method** (Sneath, 1957; Sokal y Sneath, 1963; Johnson, 1967)
- ▶ Dados dos clusters  $C_i$  y  $C_j$  la distancia entre ellos es la disimilitud más pequeña entre uno sus miembros, i.e.

$$d(C_i, C_j) = \min \left\{ d_{rs} : r \in C_i, s \in C_j \right\}$$

- ▶ **Algoritmo**
  - Buscar la disimilitud más pequeña entre clusters
  - Recalcular la matriz de disimilitudes

# Vecino más lejano

- ▶ También conocido como **complete linkage method** (Sokal y Sneath, 1963; McQuitty, 1964)
- ▶ Dados dos clusters  $C_i$  y  $C_j$  la distancia entre ellos es la disimilitud más grande entre uno sus miembros, i.e.

$$d(C_i, C_j) = \max \left\{ d_{rs} : r \in C_i, s \in C_j \right\}$$

- ▶ **Algoritmo**
  - Buscar la disimilitud más pequeña entre clusters
  - Recalcular la matriz de disimilitudes

# Centroide

- ▶ Dados dos clusters  $C_i$  y  $C_j$  se define la distancia entre ellos como la “distancia” entre sus centroides (Sokal y Michener, 1958; King, 1966, 1967)

$$\bar{X}_i = \sum_{n \in C_i} \frac{X_n}{n_i} \quad \bar{X}_j = \sum_{m \in C_j} \frac{X_m}{n_j} \quad \rightarrow \quad d(C_i, C_j) = \delta(\bar{X}_i, \bar{X}_j)$$

- ▶ **Algoritmo**

- Buscar la disimilitud más pequeña entre clusters
- Recalcular el centroide

$$\bar{X}_{C_i \cup C_j} = \frac{n_i \bar{X}_i + n_j \bar{X}_j}{n_i + n_j}$$

# Ward

- ▶ También conocido como **incremental sum of squares method** (Wishart, 1969a)

basado en la idea de Ward (1963)

- ▶ **Algoritmo**

- Unir los clusters que minimicen

$$I_{C_i C_j} = \sum_{k \in C_i \cup C_j} ||\mathbf{X}_k - \bar{\mathbf{X}}||^2 - \left[ \sum_{n \in C_i} ||\mathbf{X}_n - \bar{\mathbf{X}}_i||^2 + \sum_{m \in C_j} ||\mathbf{X}_m - \bar{\mathbf{X}}_j||^2 \right] = \frac{n_i n_j}{n_i + n_j} ||\bar{\mathbf{X}}_i - \bar{\mathbf{X}}_j||^2$$

- En particular para dos observaciones  $r, s$

$$I_{rs} = \frac{1}{2} ||\mathbf{X}_r - \mathbf{X}_s||^2 = \frac{1}{2} d_{rs}^2$$

# Promedio

- ▶ También conocido como **group average method** (Sokal y Michener, 1958; McQuitty, 1964; Lance y Williams, 1966)
- ▶ Dados dos clusters  $C_i$  y  $C_j$  la distancia entre ellos se define como el promedio de las distancias de sus miembros

$$d(C_i, C_j) = \frac{1}{n_i n_j} \sum_{n \in C_i} \sum_{m \in C_j} d_{nm}$$

# Lance y Williams

- ▶ También conocido como **Lance and Williams Flexible Method** (Lance y Williams, 1967a)

- ▶ Dados tres clusters  $C_i$ ,  $C_j$  y  $C_k$  definimos la distancia de  $C_k$  y  $C_i \cup C_j$  como:

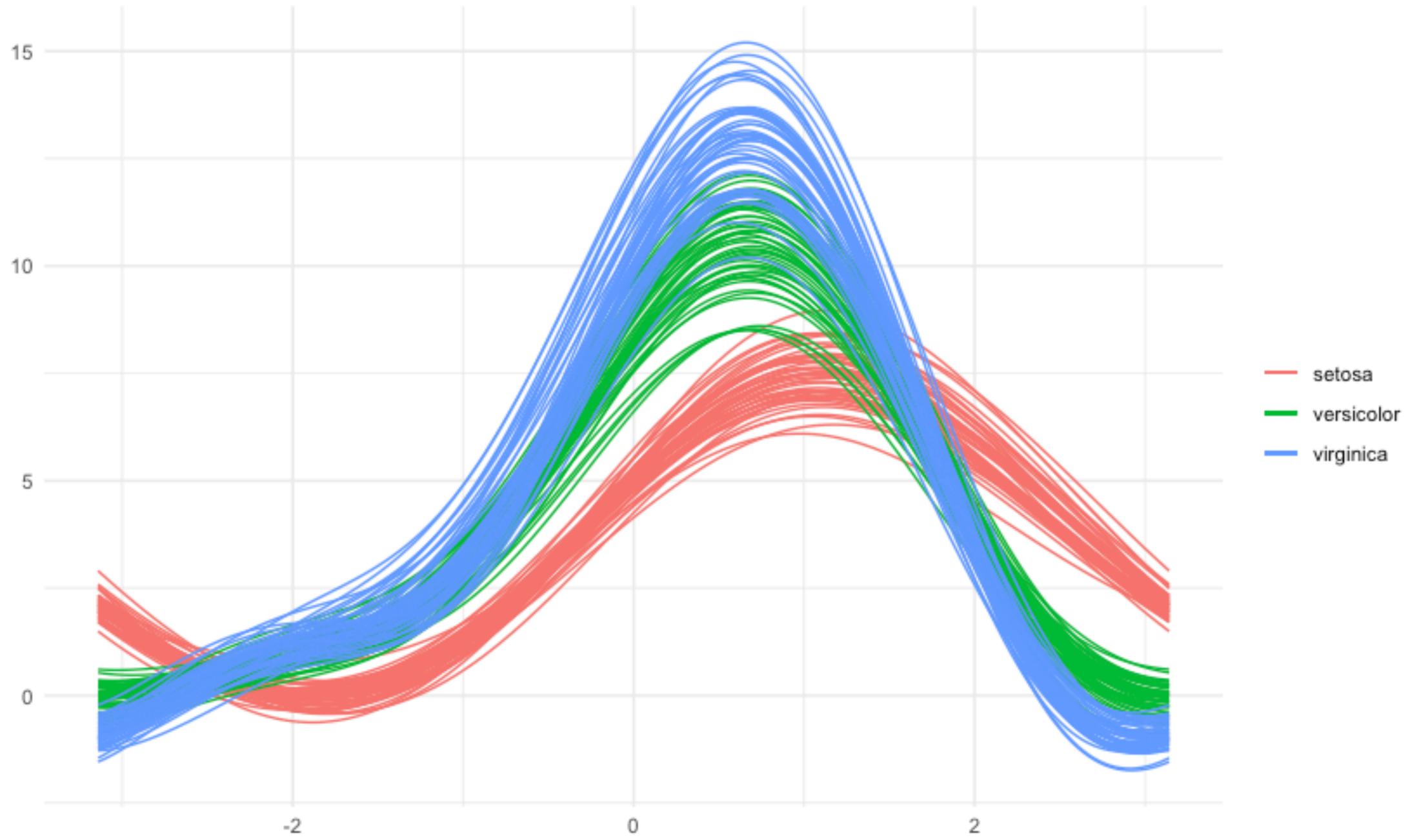
$$d\left(C_k, C_i \cup C_j\right) = \alpha_1 d(C_k, C_i) + \alpha_2 d(C_k, C_j) + \beta d(C_i, C_j) + \gamma |d(C_k, C_i) - d(C_k, C_j)|$$

- ▶ **Casos particulares:** vecino más cercano, vecino más lejano, centroide, Ward y promedio
- ▶ Lance y Williams sugieren  $\alpha_1 = \alpha_2$ ,  $\beta < 1$ ,  $\alpha_1 + \alpha_2 + \beta = 1$ ,  $\gamma = 0$

# Implementación

- ▶ En **R**: librería **cluster**
- ▶ Para un clustering aglomerativo se usa la función **agnes** y recibe como parámetros:
  - **x**: datos (matriz o data frame) o matriz de disimilitudes
  - **diss**: booleano indicando si **x** es una matriz de disimilitudes
  - **metric**: la métrica para calcular las disimilitudes de **x**
  - **stand**: booleano indicando si se deben estandarizar los datos
  - **method**: la liga a utilizar par el clustering

# Ejemplo Iris



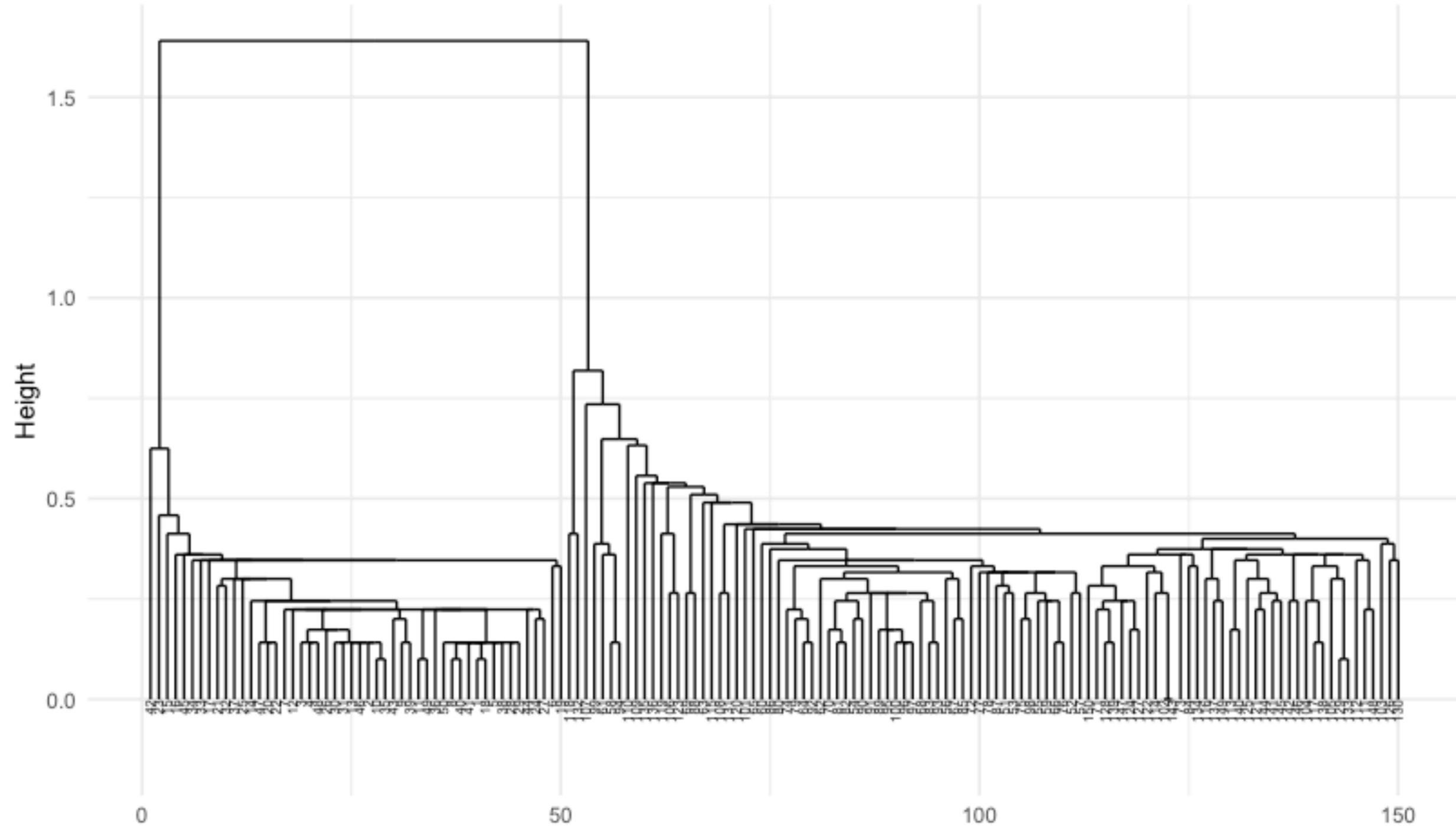
# Ejemplo Iris

- ▶ Estos métodos son para datos de los cuales no se sabe el grupo
- ▶ Iris tiene etiquetas y **solo** lo utilizamos para ejemplificar los métodos y ver como algunas ligas pueden tener un mejor performance
- ▶ Si en la vida real ya conocen los grupos no es necesario hacer clustering a menos de que existan dudas en los grupos iniciales



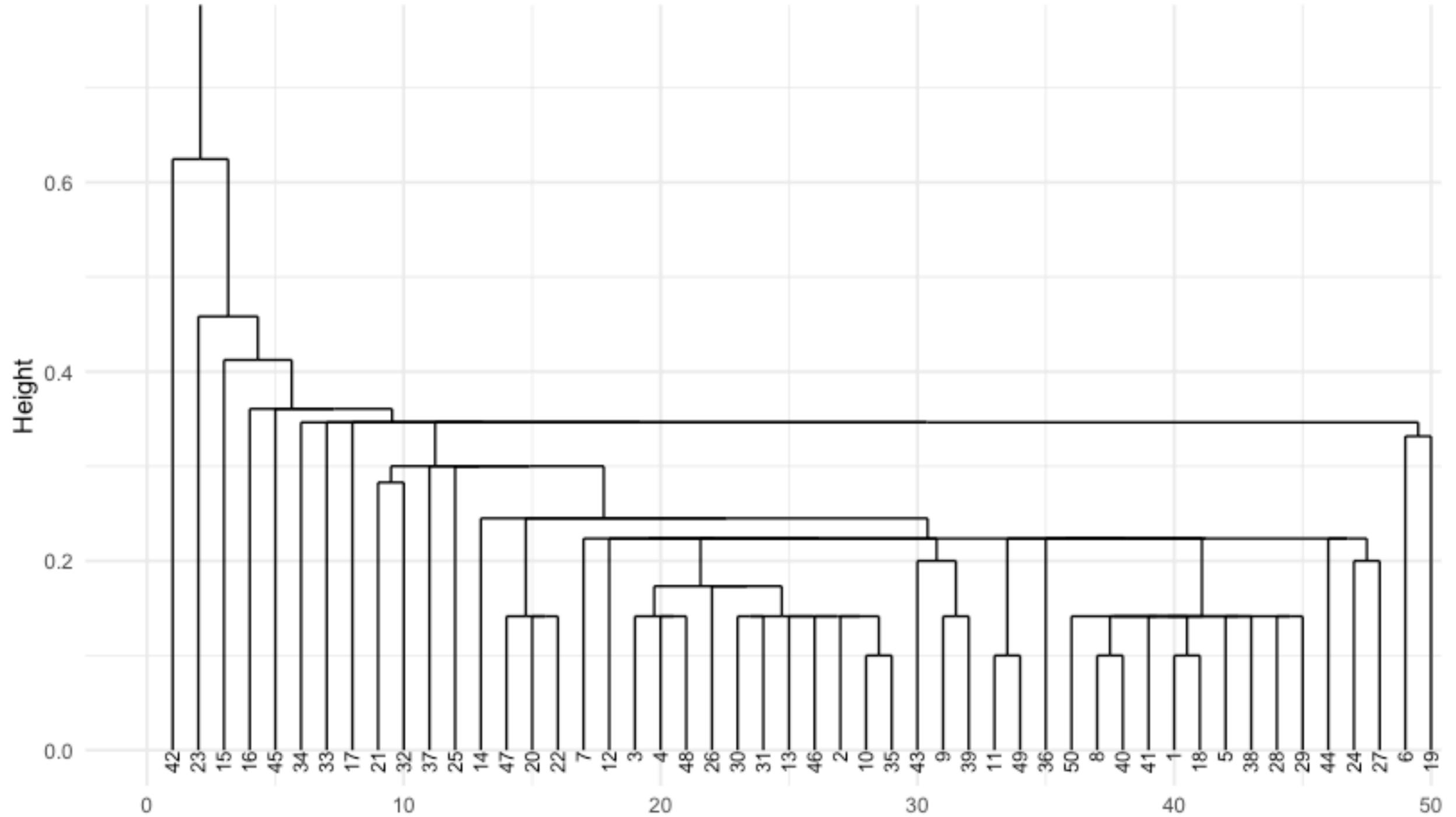
# Ejemplo Iris

- Vecino más cercano



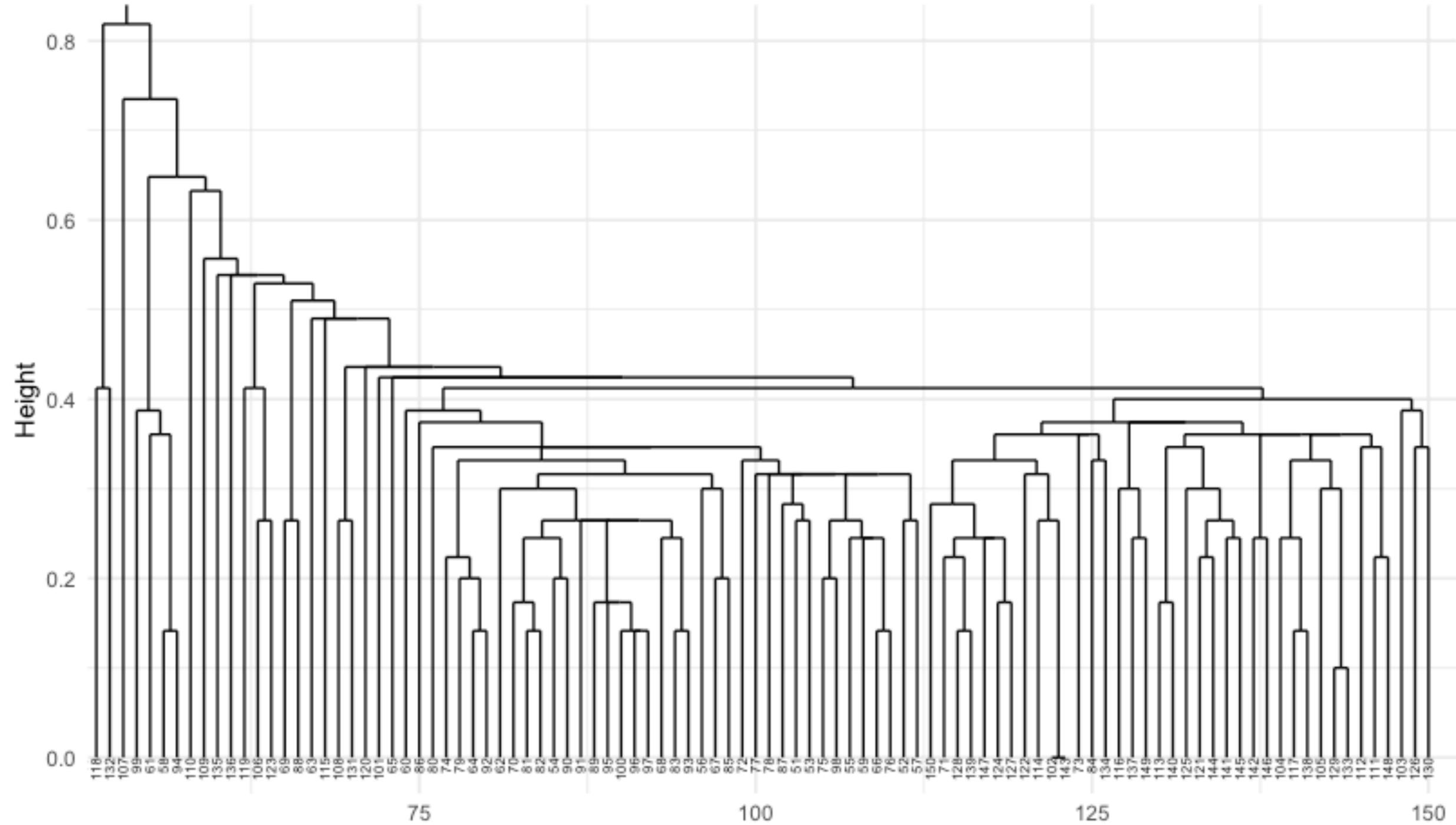
# Ejemplo Iris

- ▶ Primer grupo



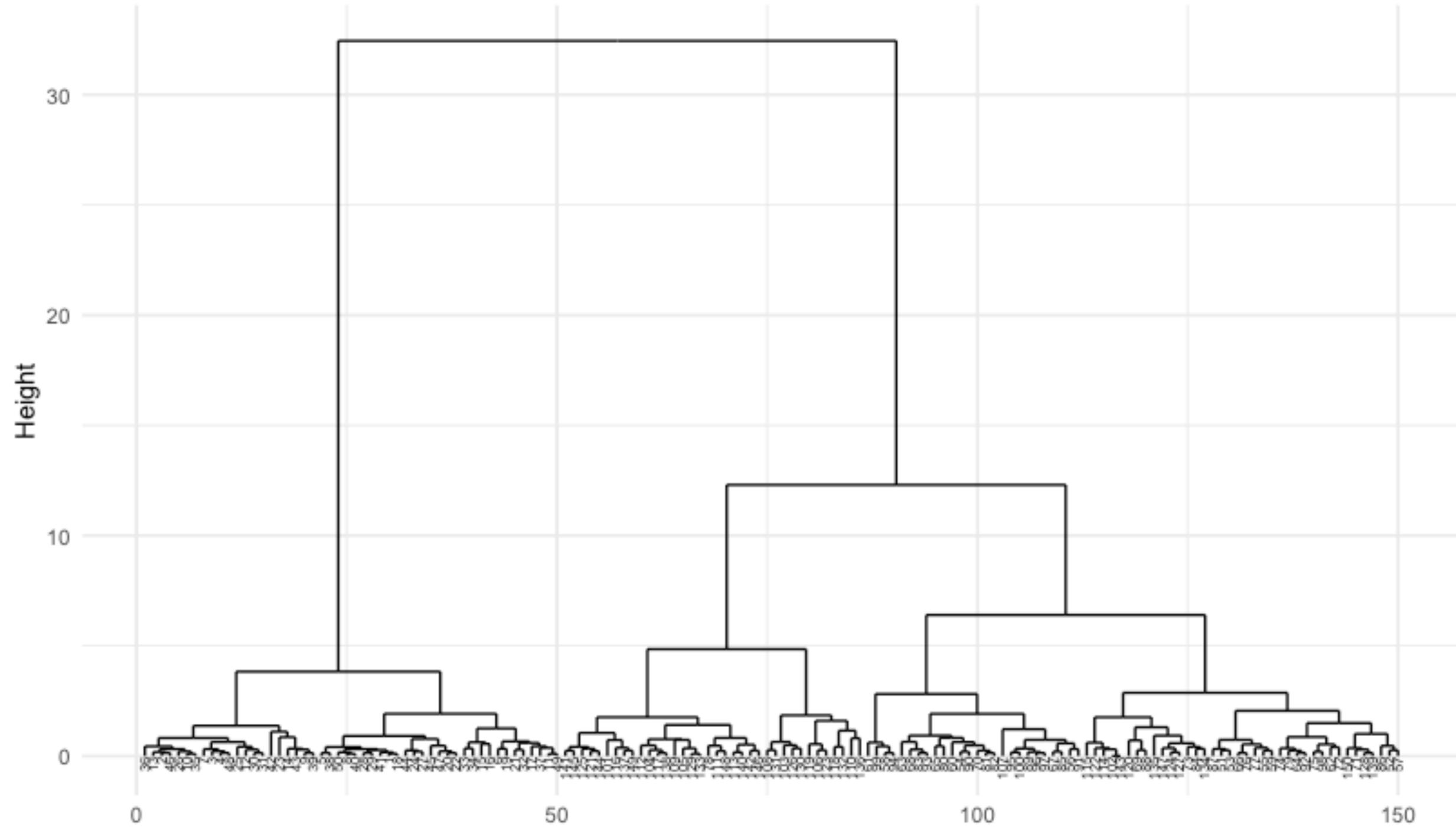
# Ejemplo Iris

- Segundo grupo



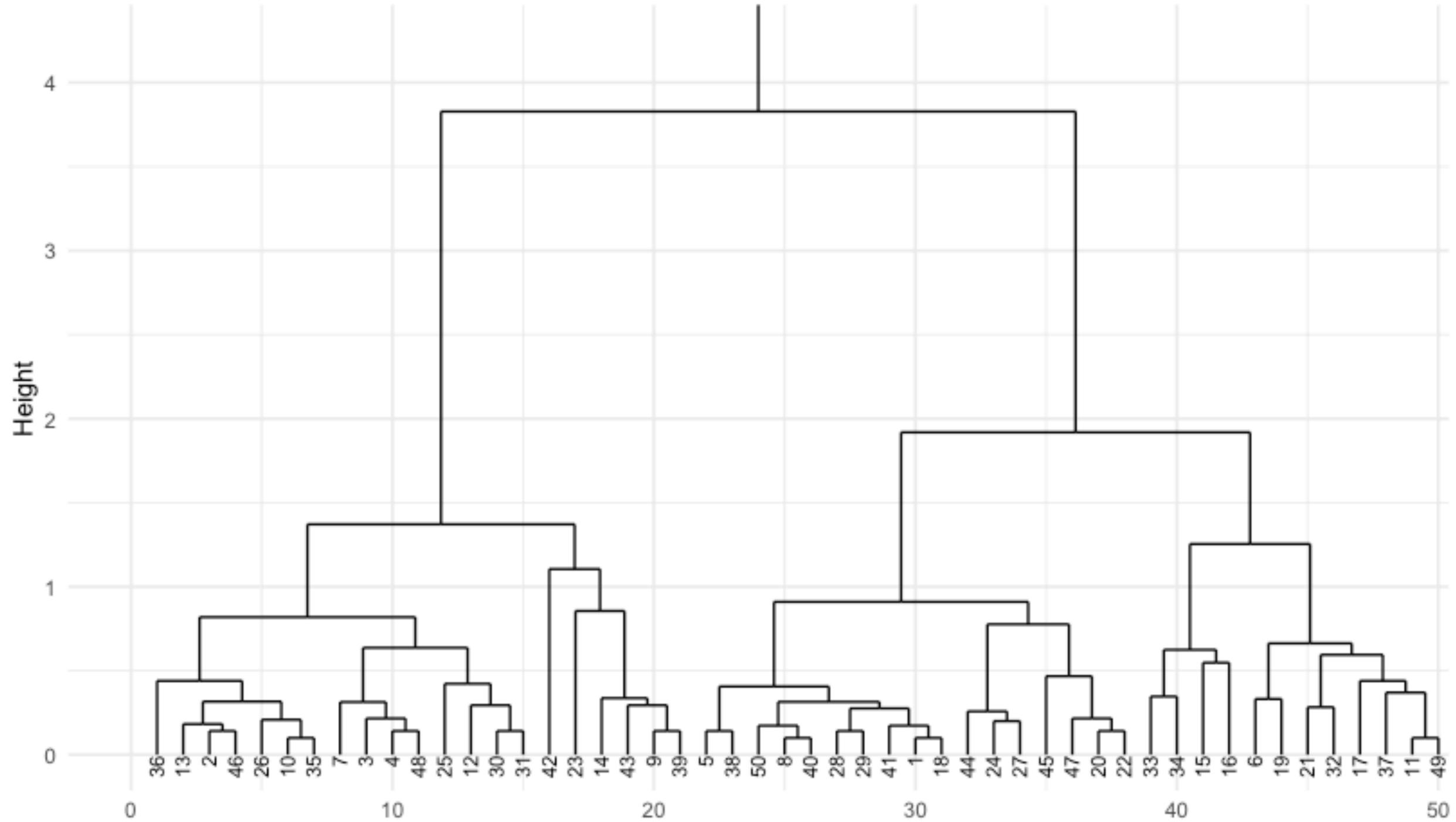
# Ejemplo Iris

- Liga Ward



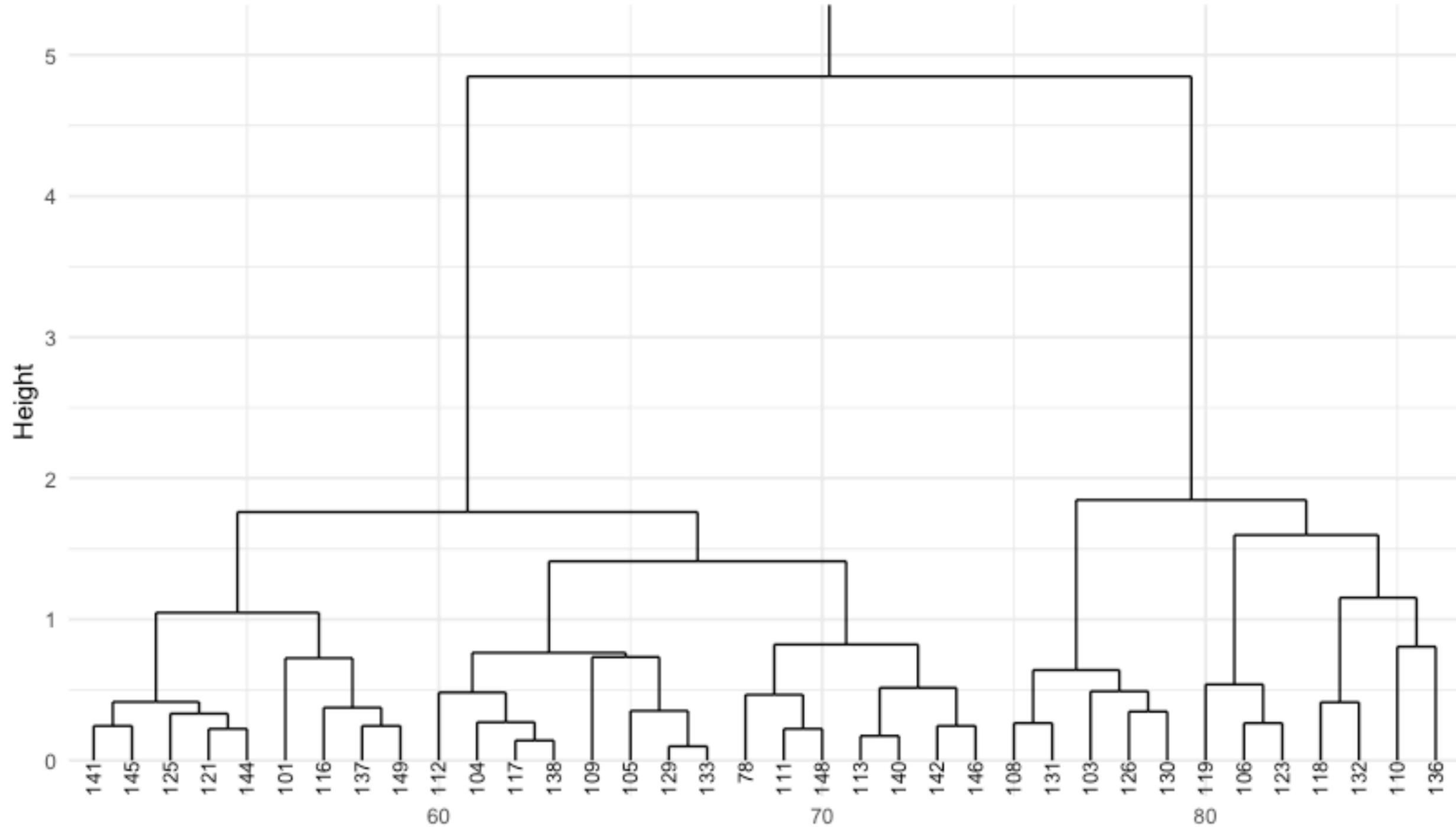
# Ejemplo Iris

- ▶ Primer grupo



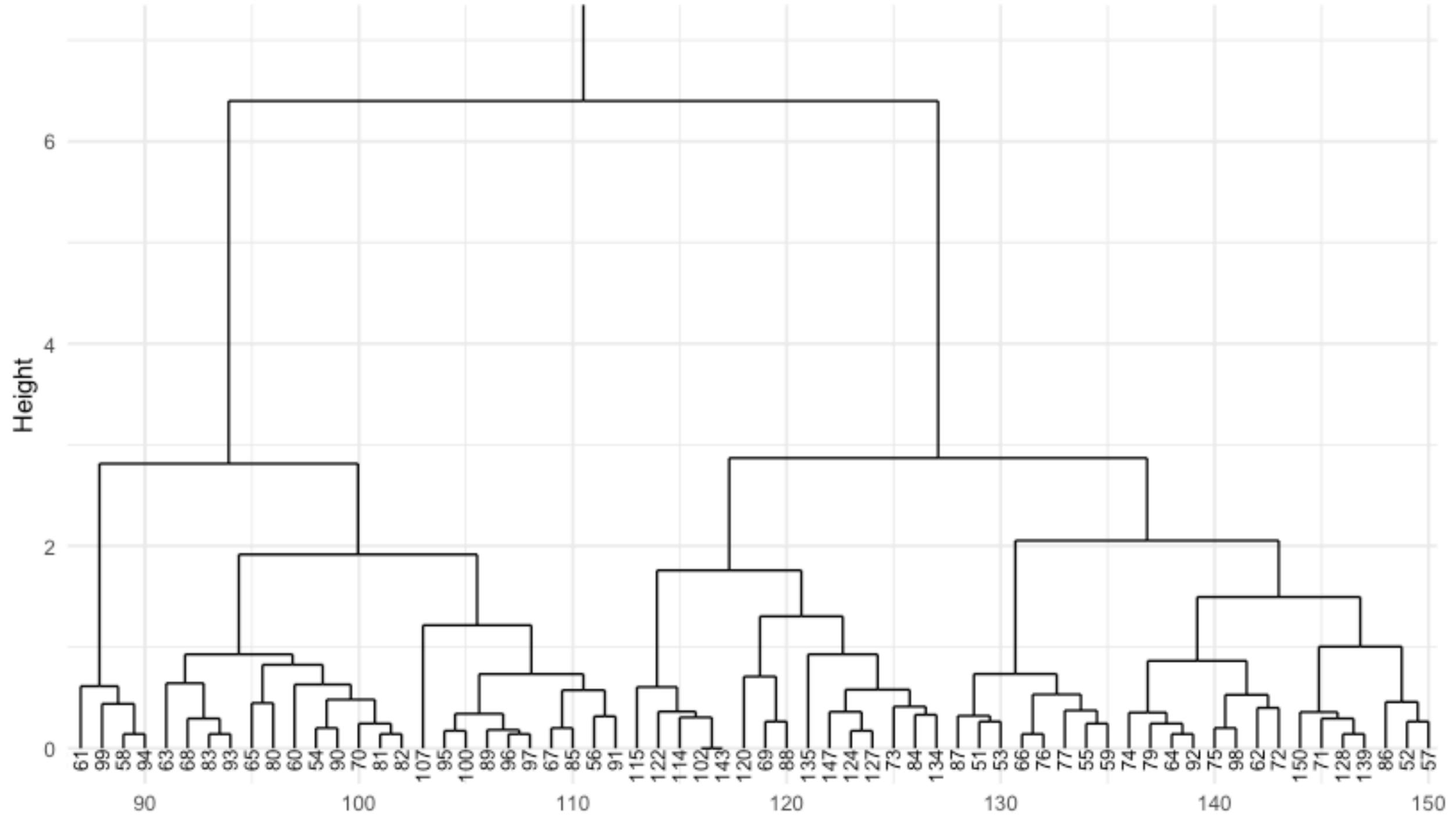
# Ejemplo Iris

- Segundo grupo



# Ejemplo Iris

- Tercer grupo



# Silhouette

▸ Medida de similitud de un objeto y el cluster al que pertenece comparado con el resto (Rousseeuw, 1987).

▸ Construcción para el  $i$ -ésimo objeto en el cluster  $A$ :

- Obtener la disimilitud promedio de su cluster  $a(i)$

- Obtener el mínimo de las disimilitudes promedio de los otros clusters, i.e.,

$b(i) = \min_{C \neq A} \{d(i, C)\}$  (dicho cluster es la segunda mejor opción)

- Definimos la silhouette como:

$$s(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}}$$

# Silhouette

- ▶  $-1 \leq s(i) \leq 1$  por lo que:
  - Si  $s(i) > 0$  el objeto está bien clasificado
  - Si  $s(i) = 0$  el objeto está a la misma distancia de  $A$  y de  $B$
  - Si  $s(i) < 0$  el objeto está mal clasificado
- ▶ Podemos crear una gráfica poniendo los silhouettes ordenados por cada cluster, en **R** usar la función `silhouette()`
- ▶ Proporciona una forma de medir que tanta estructura hemos descubierto usando el promedio de las silhouettes  $\bar{s}(k)$  y una posible forma de elegir  $k$  con el silhouette coefficient  $SC = \max_k \{\bar{s}(k)\}$

# Silhouette

▸ (Posible interpretación) Kaufman (1990) proporciona la siguiente tabla basado en su experiencia:

- Si  $SC \in (.70,1]$  se ha encontrado una fuerte estructura de clustering
- Si  $SC \in (.5,.7]$  se ha encontrado una estructura razonable
- Si  $SC \in (.25,.5]$  la estructura es débil y se debe considerar otro método
- Si  $SC \leq .25$  no se encontró una estructura sustancial

# Ejemplo Iris + AGNES (single)

## Silhouette

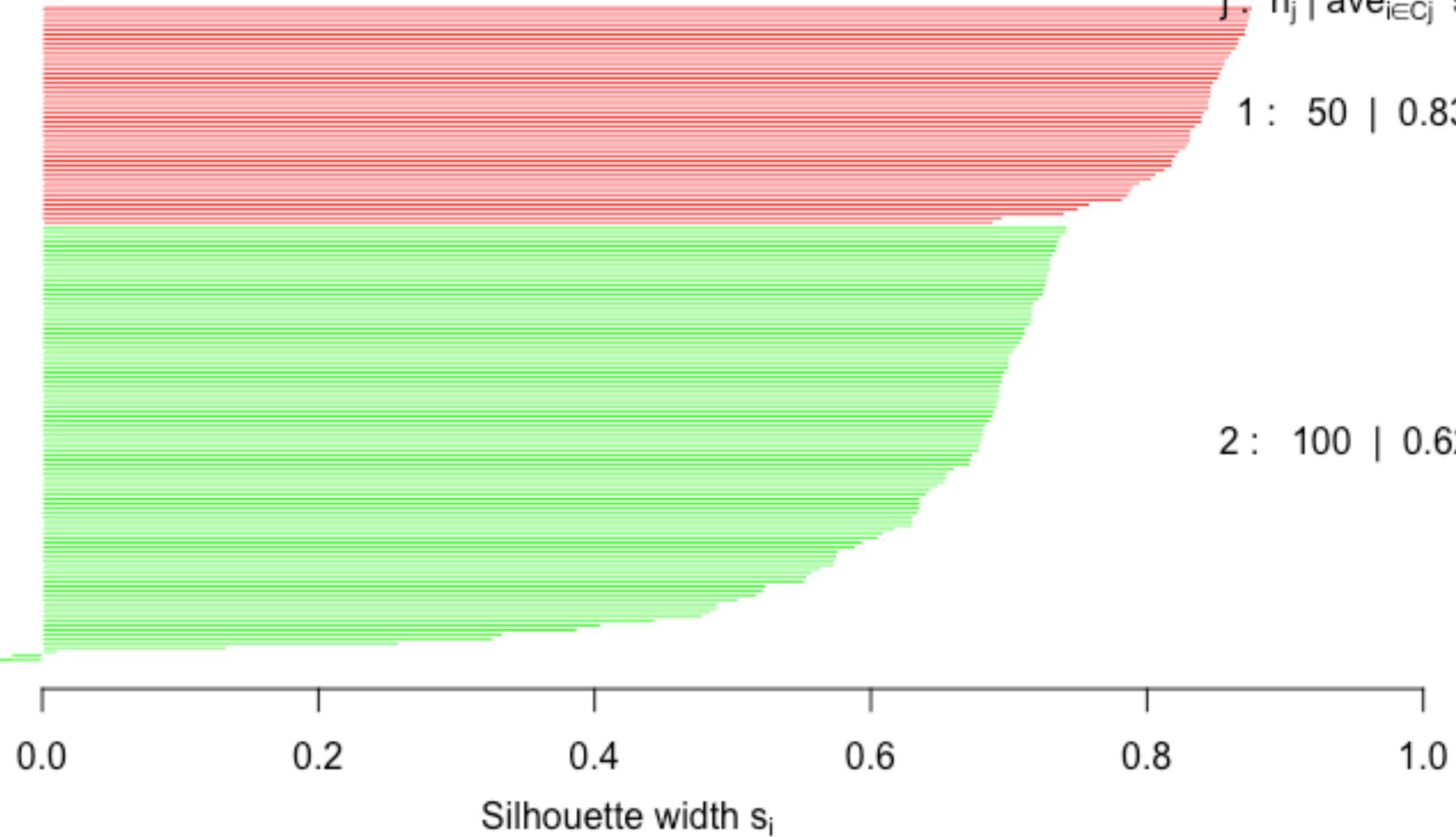
n = 150

2 clusters  $C_j$

$j : n_j \mid \text{ave}_{i \in C_j} s_i$

1 : 50 | 0.83

2 : 100 | 0.62



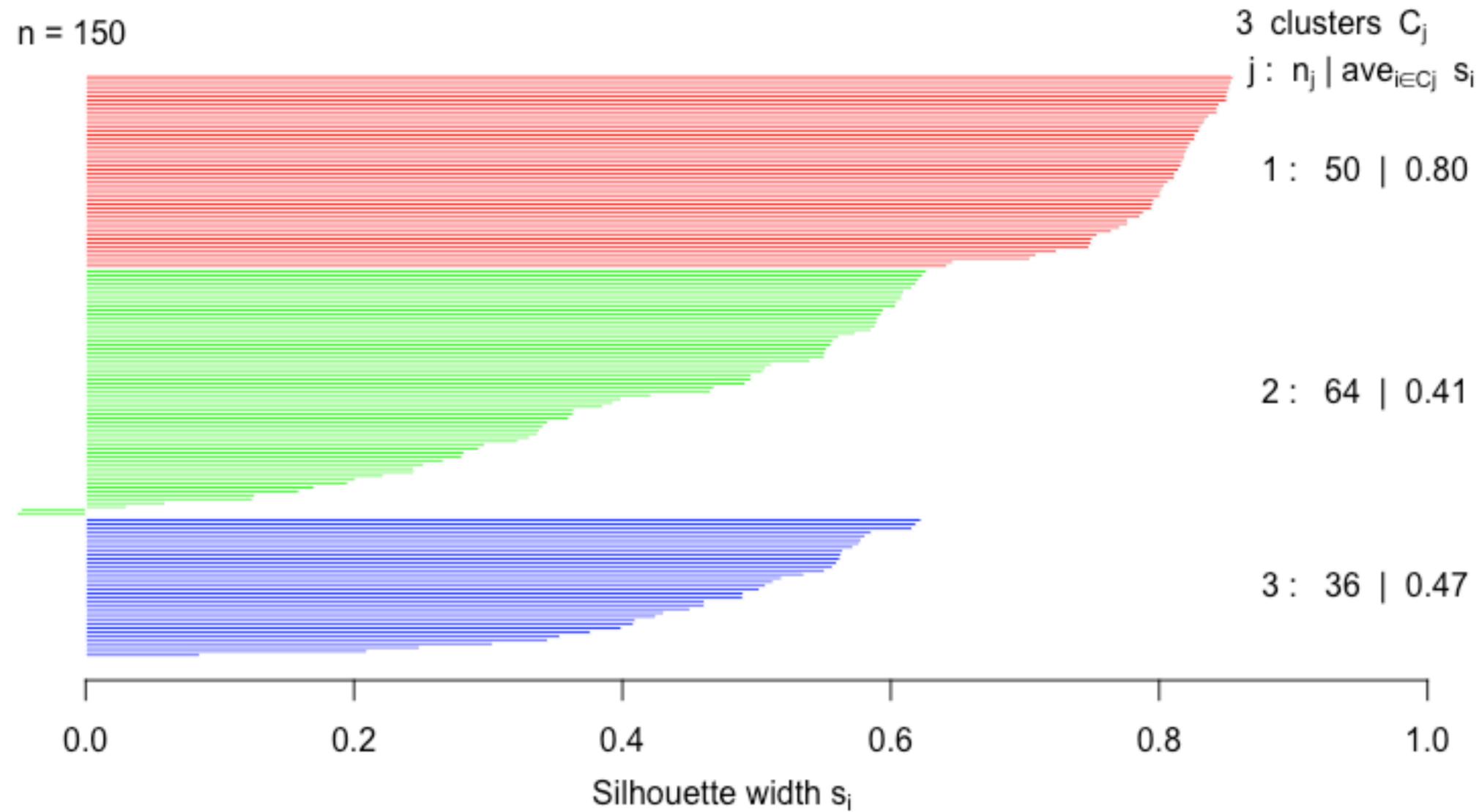
Average silhouette width : 0.69

Obs.	Cluster	Alternativa	Silhouette
58	2	1	-0.0221145
99	2	1	-0.15177853

# Ejemplo Iris + AGNES (Ward)

## Silhouette

n = 150



Average silhouette width : 0.55

Obs.	Cluster	Alternativa	Silhouette
53	2	3	-0.05092527
135	2	3	-0.04842929

# Métodos jerárquicos divisivos (DIANA)

# Motivación

- ▶ **Idea**

Empezar con un cluster de tamaño  $n$  e irlo dividiendo.

- ▶ **Ventajas** (Williams y Lance, 1977)

- El proceso empieza con el contenido máximo de información.
- La división no tiene que continuar hasta tener  $n$  clusters.

- ▶ **Restricción**

- $2^{n-1} - 1$  formas de separar  $n$  objetos en 2 grupos ... imposible analizar todos los casos

# Mono-variables

- ▶ **Idea**

La división se basa en una sola variable

- ▶ **Problemas**

- Sensible a outliers
- Difícil de adaptar con una mezcla de variables cuantitativas y cualitativas
- Errores frecuentes de clasificación

# Mono-variables

- ▶ Si todas las variables son dicotómicas:
  - Dividimos las observaciones en dos grupos dependiendo si tienen el atributo  $A$
- ▶ **¿Cómo elegimos a  $A$ ?**
  - Maximice alguna medida de distancia entre dos grupos e.g.: estadístico  $\chi^2$

Var. r \ Var. s	Presente	Ausente	Total
Presente	a	b	a+b
Ausente	c	d	c+d
Total	a+c	b+d	a+b+c+d

$$\chi_{rs}^2 = \frac{n(ad - bc)^2}{(a + b)(a + c)(b + d)(c + d)}$$

Elegimos  $A$  como la que maximice:  $\sum_{j:j \neq A} \chi_{jA}^2$

# Mono-variables

- ▶ Para variables cuantitativas:

- Dividir de tal forma que se minimice la suma de cuadrados dentro del grupo (**within-group**) o maximizar la suma de cuadrados entre grupos (**between-group**)

$$B = n_1 (\bar{x}_1 - \bar{x})^2 + n_2 (\bar{x}_2 - \bar{x})^2 = n_1 \bar{x}_1^2 + n_2 \bar{x}_2^2 - n \bar{x}^2$$

- ▶ Para no considerar todas las posibles divisiones se sugiere:

- Ordenar los datos y hacer la división en la  $r$  donde se maximice

$$R = r \bar{x}_1^2 + (n - r) \bar{x}_2^2$$

# Splinter

- ▶ **Objetivo**

La división se basa elegir en cada paso a la observación más disimilar en promedio

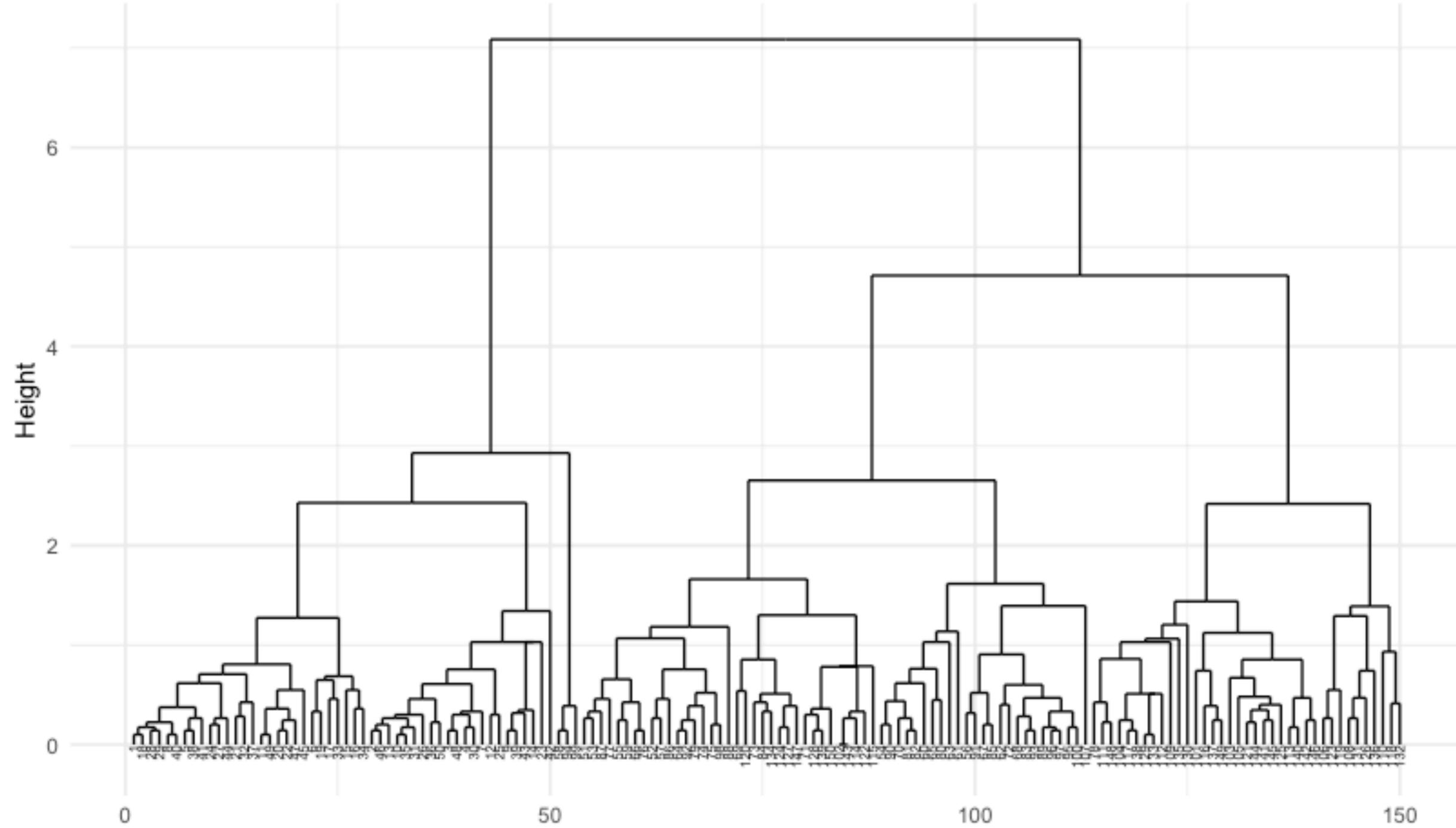
- ▶ **Algoritmo**

- Seleccionar el cluster más grande
- Buscar la observación más disimilar en promedio
- Empezar un nuevo grupo con esta observación
- Reagrupar las observaciones dependiendo su disimilitud entre los miembros del grupo antiguo y el nuevo

# Implementación

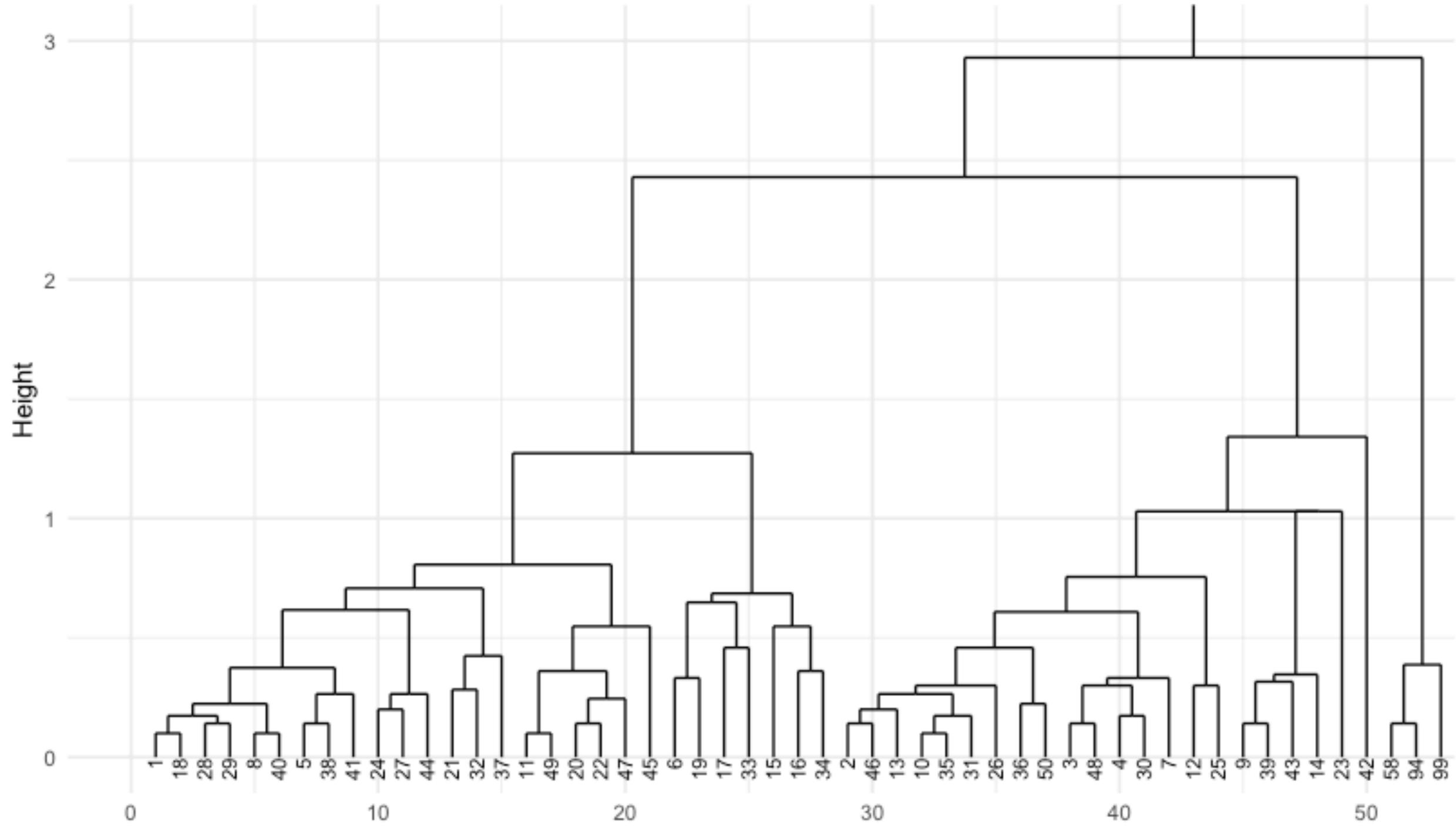
- ▶ En **R**: librería **cluster**
- ▶ Para un clustering aglomerativo se usa la función **diana** y recibe como parámetros:
  - **x**: datos (matriz o data frame) o matriz de disimilitudes
  - **diss**: booleano indicando si **x** es una matriz de disimilitudes
  - **metric**: la métrica para calcular las disimilitudes de **x**
  - **stand**: booleano indicando si se deben estandarizar los datos
  - **stop.at.k**: el valor en el cual se debe detener la división

# Ejemplo Iris



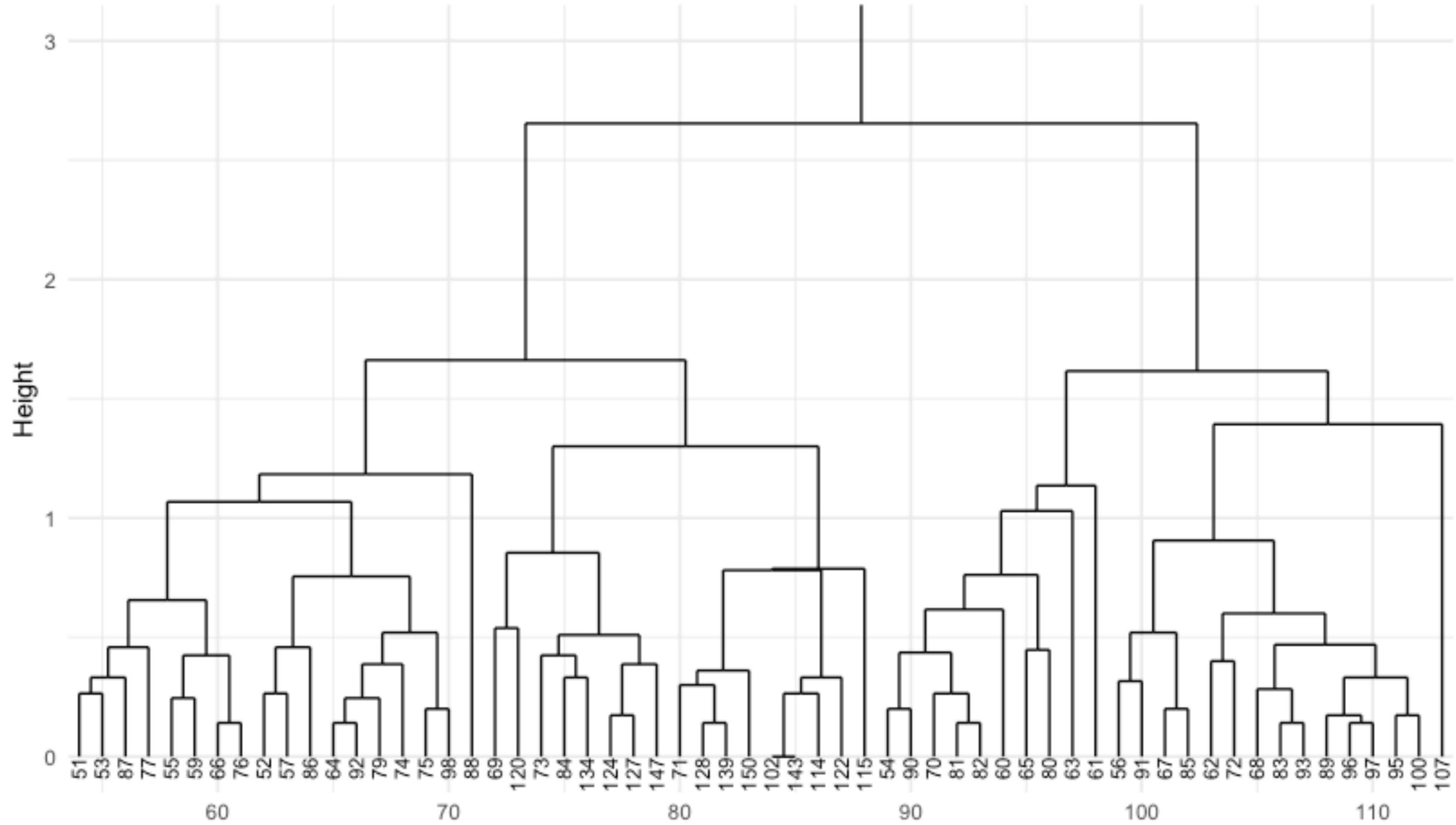
# Ejemplo Iris

- ▶ Primer grupo



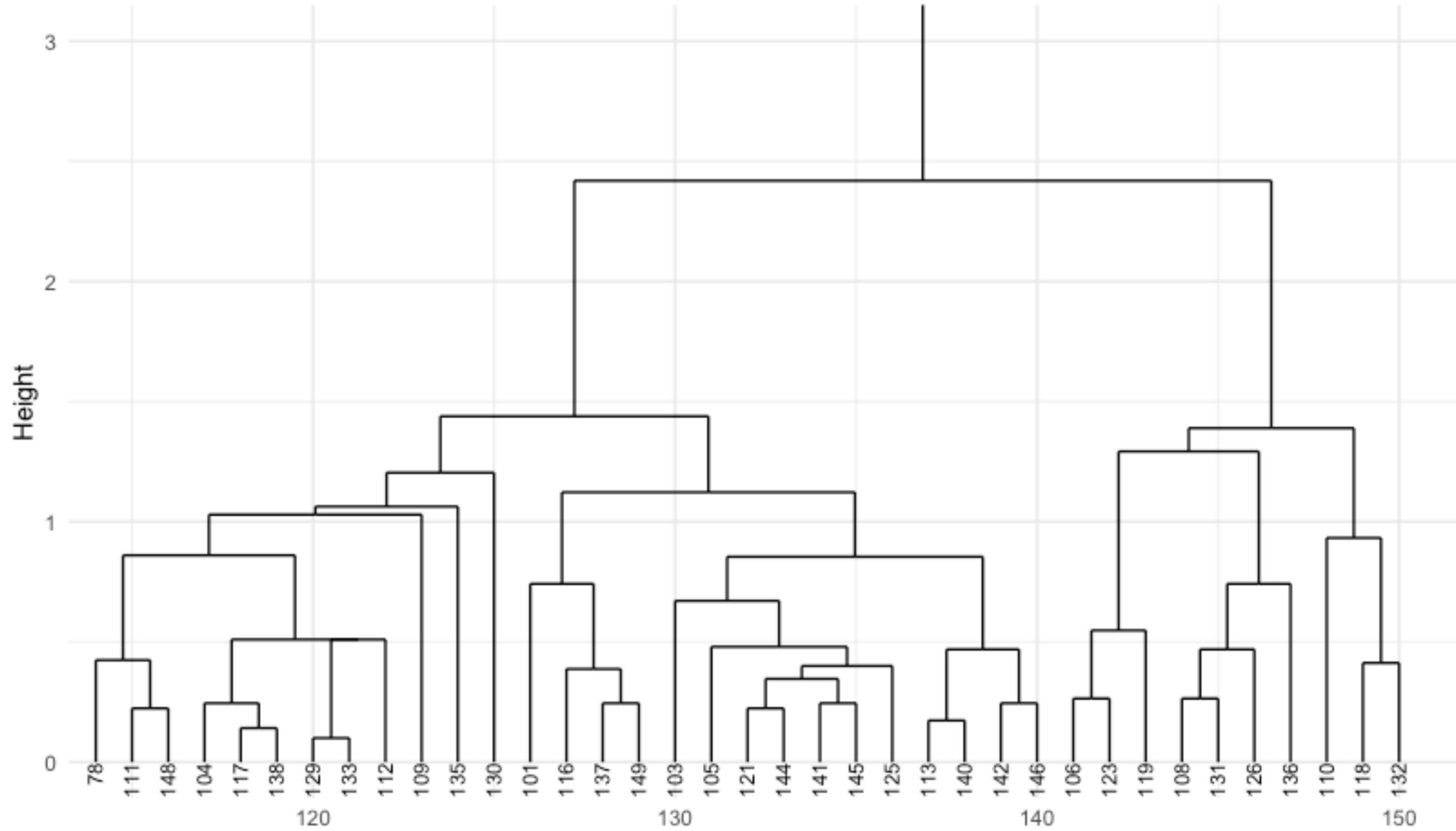
# Ejemplo Iris

- Segundo grupo



# Ejemplo Iris

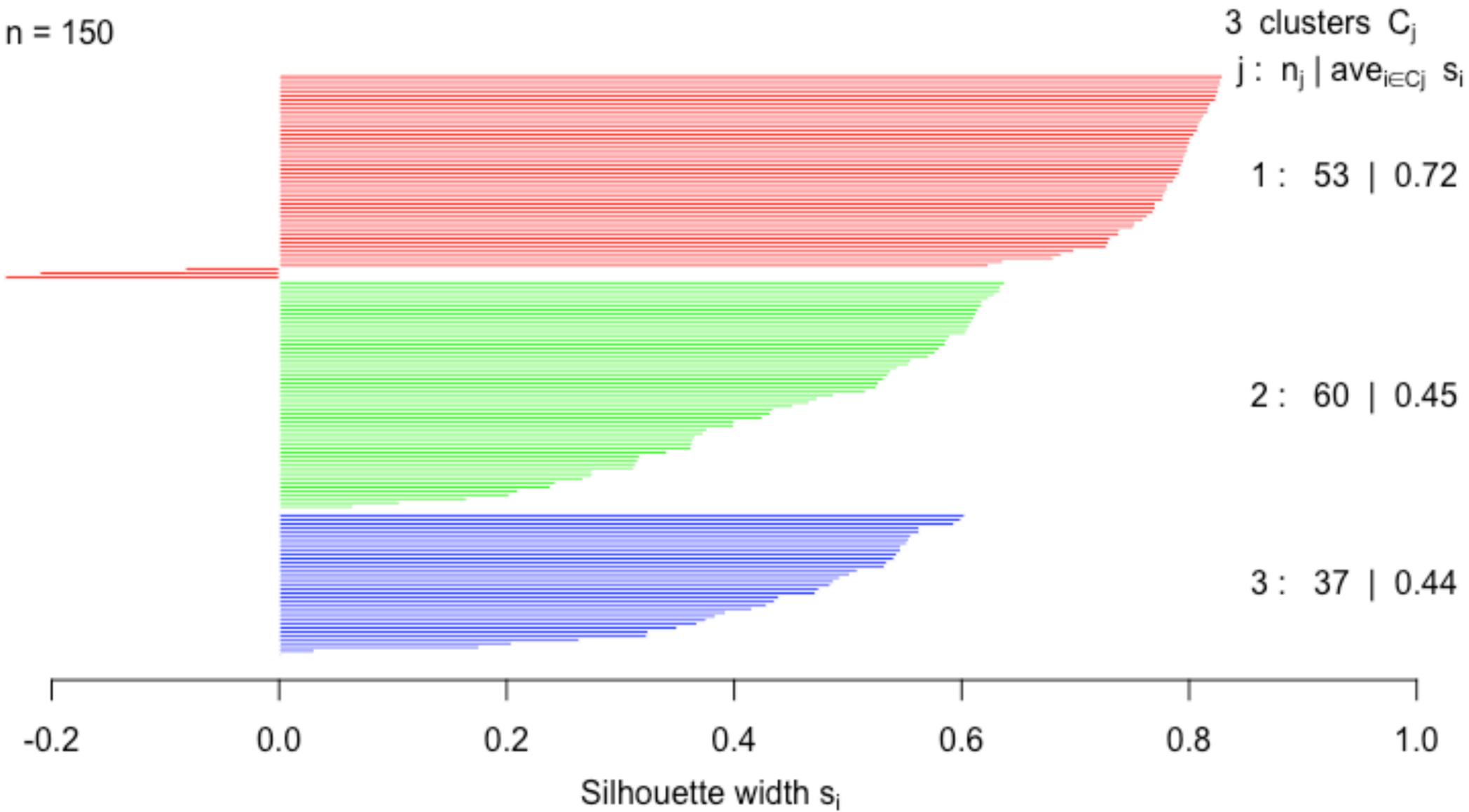
- Tercer grupo



# Ejemplo Iris + DIANA

## Silhouette

n = 150



Obs.	Cluste	Alternativa	Silhouette
58	1	2	-0.21035377
94	1	2	-0.24078366
99	1	2	-0.082634985

# Métodos de particiones

# Motivación

## ¿De qué va?

▸ Dado un número de clusters  $k$  se busca agrupar las  $n$  observaciones en estos clusters optimizando algún criterio.

## ▸ Dificultad

El número de formas de separar  $n$  objetos en  $k$  grupos está dado por el número de Sterling del segundo tipo:

$$S(n, k) = \left\{ \begin{matrix} n \\ k \end{matrix} \right\}$$

- Por ejemplo:  $S(16, 8) = 2,141,764,053$  (Imposible considerar todas las particiones)

# Algoritmo

## ¿ Cómo escoger $k$ ?

- Usar método aglomerativo (no es ideal)
- Usar algún modelo que permita reasignar las observaciones

## Algoritmo

1. Seleccionar  $k$  observaciones como los centroides de los clusters
2. Asignar el resto de las observaciones al cluster más cercano
3. Actualizar el centroide a cada paso (e.g.  $k$ -means) o hasta el final
4. Buscar objetos mal asignados y reasignar
5. Repetir hasta optimizar el criterio

# Criterios de clustering

- ▶ Varios criterios han sido propuestos basados en la identidad

$$\mathbf{T} = \mathbf{W} + \mathbf{B} = \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}}) (\mathbf{x}_i - \bar{\mathbf{x}})^T$$

donde:

- **W** es la matriz de variación dentro del cluster (**within-cluster**).
  - **B** es la matriz de variación entre clusters (**between-cluster**).
- 
- ▶ Se busca minimizar (alguna función) de **W** o maximizar (alguna función) **B**

# Criterios de clustering

- ▶ Minimizar  $\text{tr}(\mathbf{W})$

- Popular por su simplicidad y costo computacional
- Invariante ante transformaciones ortogonales pero no ante todas las transformaciones singulares no lineales (i.e. diferentes soluciones para los datos y los datos estandarizados)

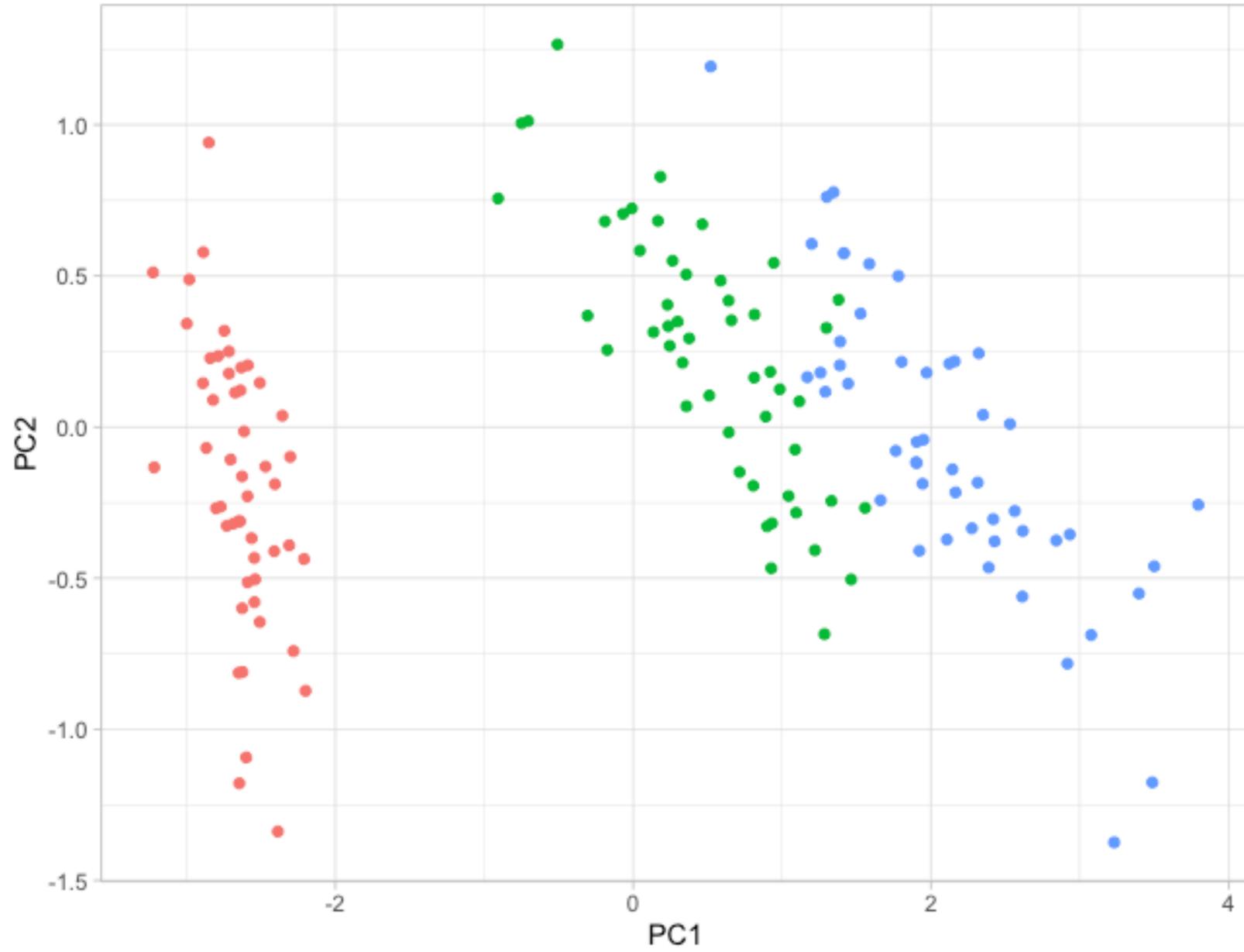
- ▶ Maximizar  $\text{tr}(\mathbf{B}\mathbf{W})^{-1}$

- No es muy confiable ya que no corrige errores en los grupos (Maronna y Jacovkis, 1974)

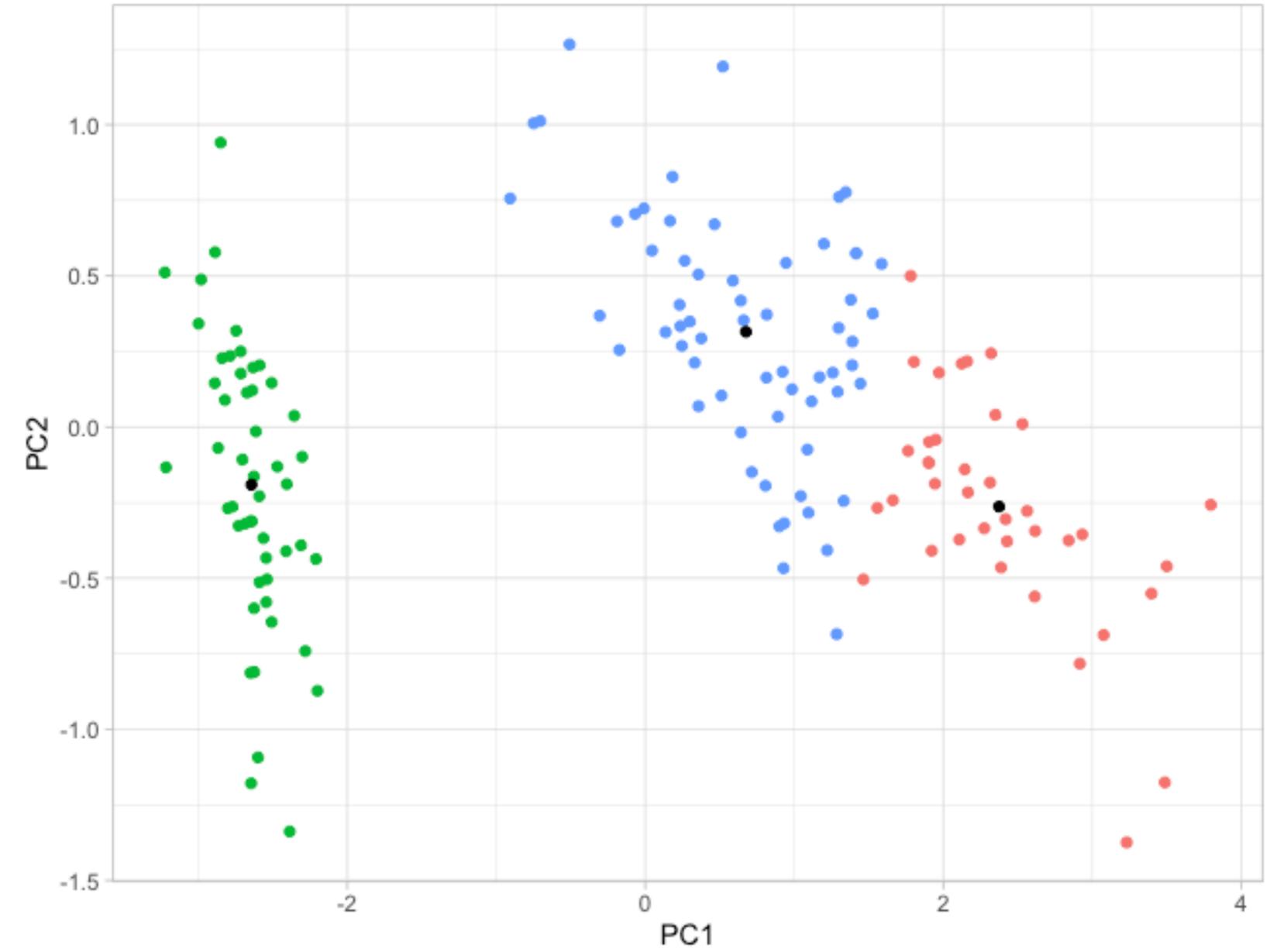
# Criterios de clustering

- ▶ Minimizar  $|W|$ 
  - Invariante ante transformaciones no singulares
  - Mayor sensibilidad a la estructura de los datos (Friedman y Rubin, 1967)
  - Puede verse influenciado por una variable que permita crear clusters bien definidos (Marriott, 1971)

# Ejemplo Iris + K-means



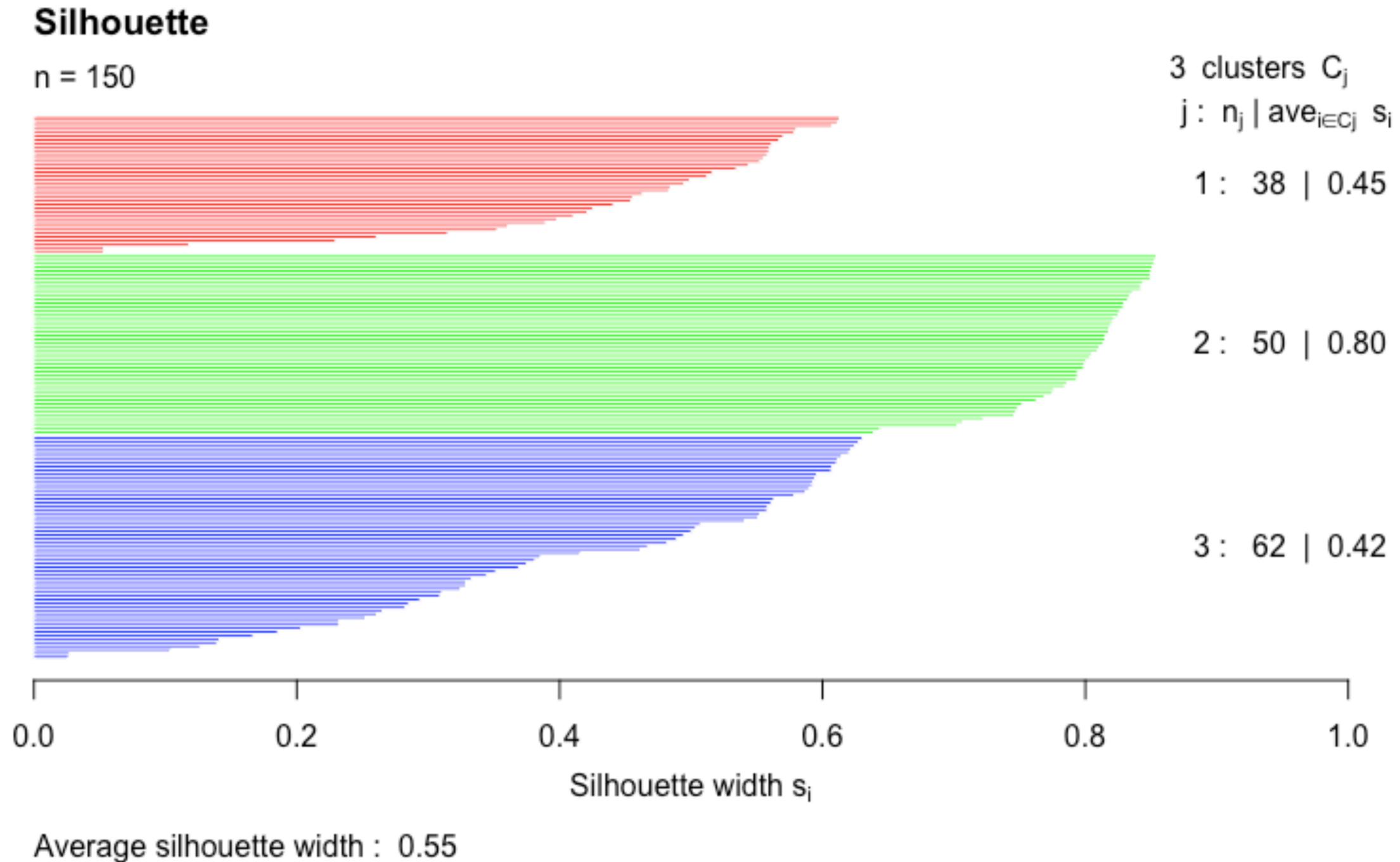
Etiquetas originales



k-means

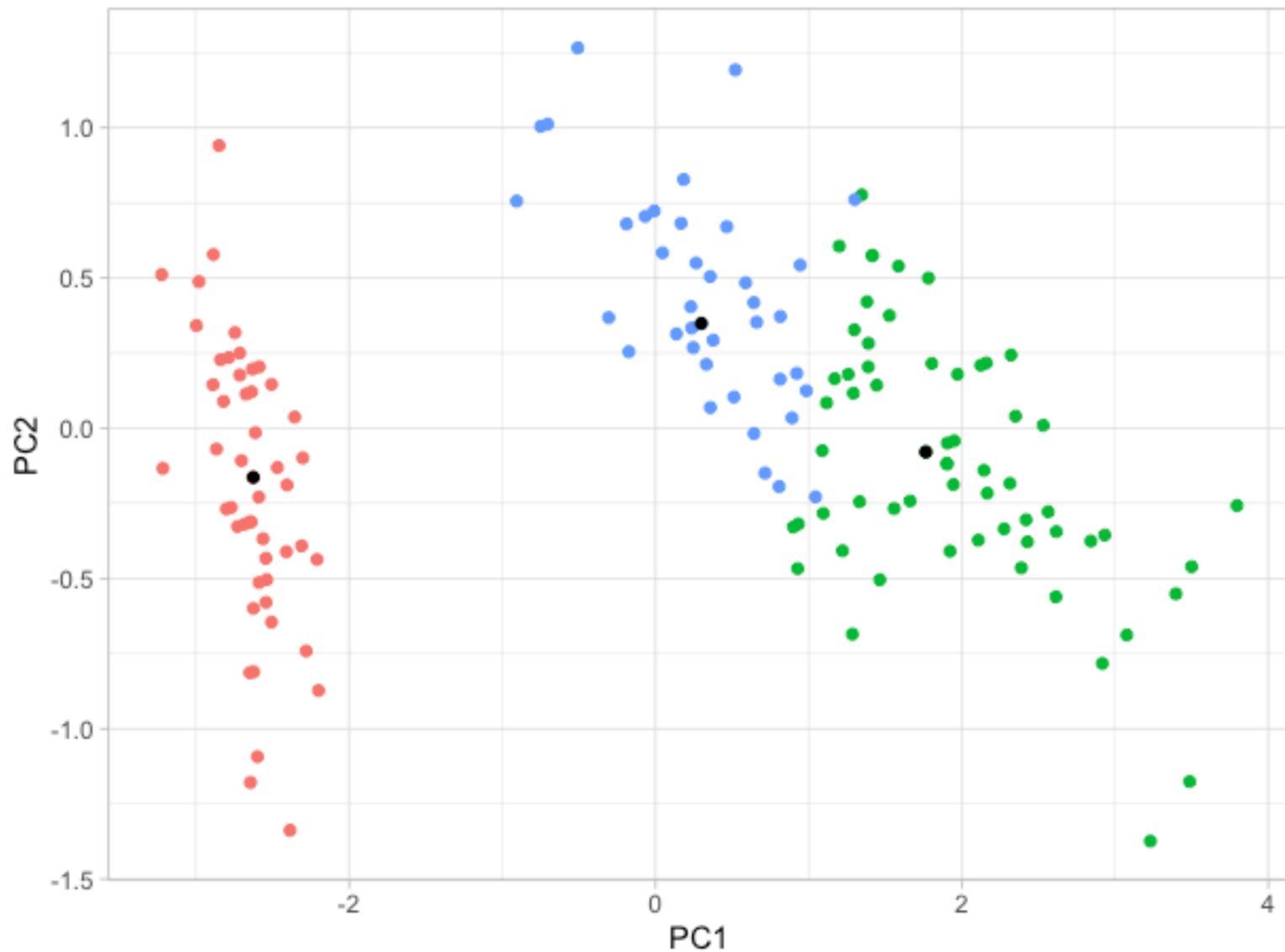
# Ejemplo Iris + K-means

- ▶ No hay indicio de observaciones mal clasificadas



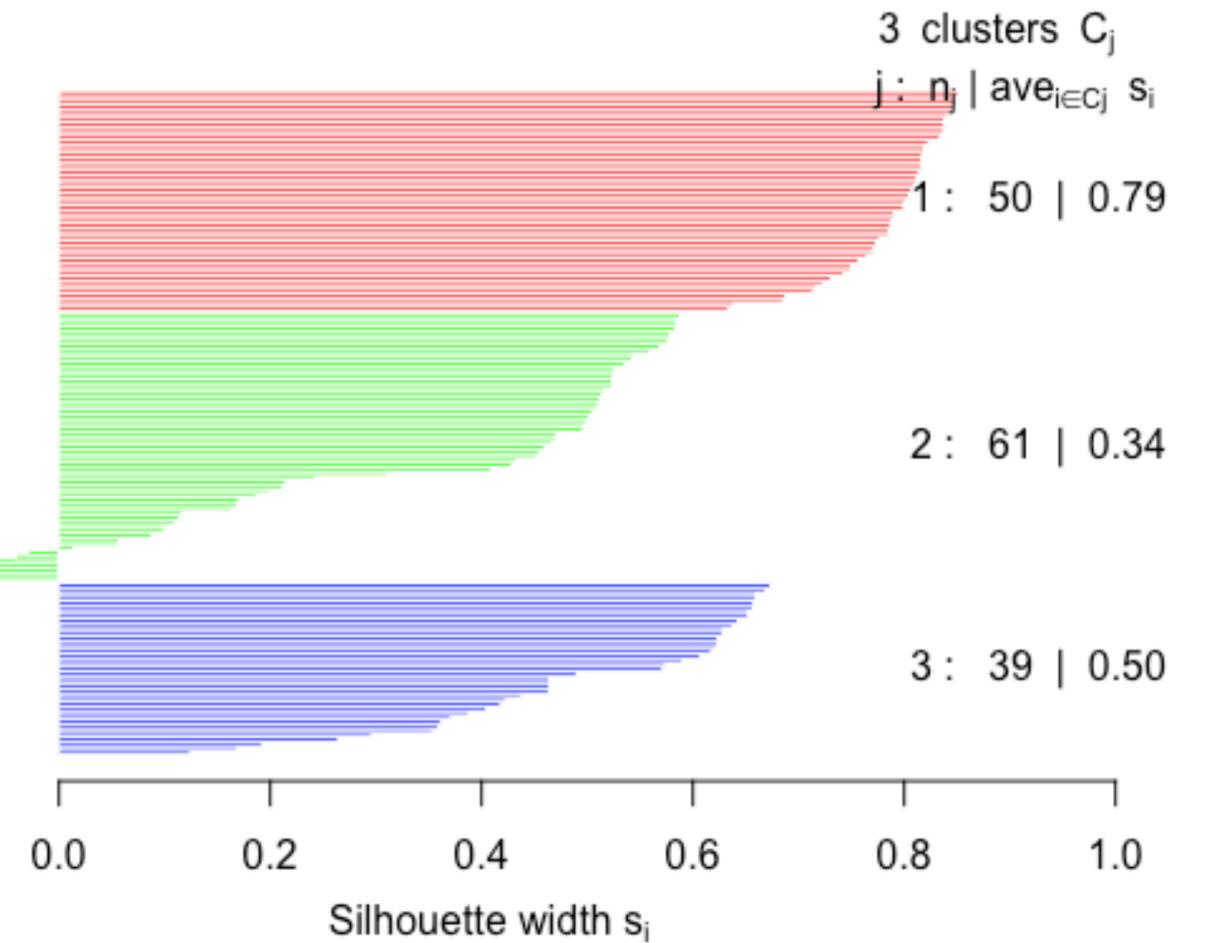
# Alternativas

- ▶ Seleccionar otra métrica (e.g. Manhattan)
- ▶ Usar el medoide en lugar de la medias (algoritmo pam() en R)



## Silhouette

n = 150



# Ejemplo Iris

- ▶ Encontramos que las observaciones mal clasificadas son:

<b>Observación</b>	<b>Cluster Asignado</b>	<b>Cluster Alternativo</b>	<b>Silhouette</b>
114	2	3	-0.028
122	2	3	-0.042
73	2	3	-0.063
52	2	3	-0.098
55	2	3	-0.103
66	2	3	-0.109
76	2	3	-0.185

# Métodos de Multi-Pertenencia

# Fuzzy clustering

- ▶ **Motivación**

En ocasiones es más significativo permitir que las observaciones pertenezcan a varios grupos.

- ▶ **Idea**

Encontrar un coeficiente de pertenencia para cada objeto,  $u_{im} \in [0,1]$  , (membership

coefficient) para cada cluster de tal forma que  $\sum_{m=1}^k u_{im} = 1$

# Algoritmo Fanny

- Algoritmo iterativo propuesto por Kaufman en 1990 (en **R** usamos `fanny()`)
- Se busca minimizar la función:

$$\sum_{m=1}^k \frac{\sum_{i,j=1}^n u_{im}^2 u_{jm}^2 d(i,j)}{2 \sum_{j=1}^n u_{jm}^2}$$

- Para medir el tipo de clustering (suave o duro) usamos el coeficiente de Dunn (1976):

$$F_k = \sum_{i=1}^n \sum_{m=1}^k \frac{u_{im}^2}{n}$$

- El mínimo de  $F_k$  se alcanza cuando hay máxima difusión (complete fuzziness) y el máximo cuando se crea una partición.

# Ejemplo Iris

- ▶ Algunos coeficientes de pertenencia

<b>Observación</b>	<b>Grupo 1</b>	<b>Grupo 2</b>	<b>Grupo 3</b>
<b>102</b>	0.32	0.22	0.45
<b>143</b>	0.08	0.16	0.75
<b>57</b>	0.09	0.65	0.25
<b>71</b>	0.07	0.44	0.47
<b>139</b>	0.06	0.72	0.20
<b>84</b>	0.05	0.69	0.25
<b>114</b>	0.05	0.75	0.18
<b>122</b>	0.11	0.61	0.27
<b>73</b>	0.14	0.28	0.56
<b>52</b>	0.09	0.63	0.27
<b>55</b>	0.07	0.65	0.27
<b>66</b>	0.11	0.62	0.26
<b>76</b>	0.05	0.69	0.25

- ▶ ¿Cómo podemos pasar a un clustering duro?

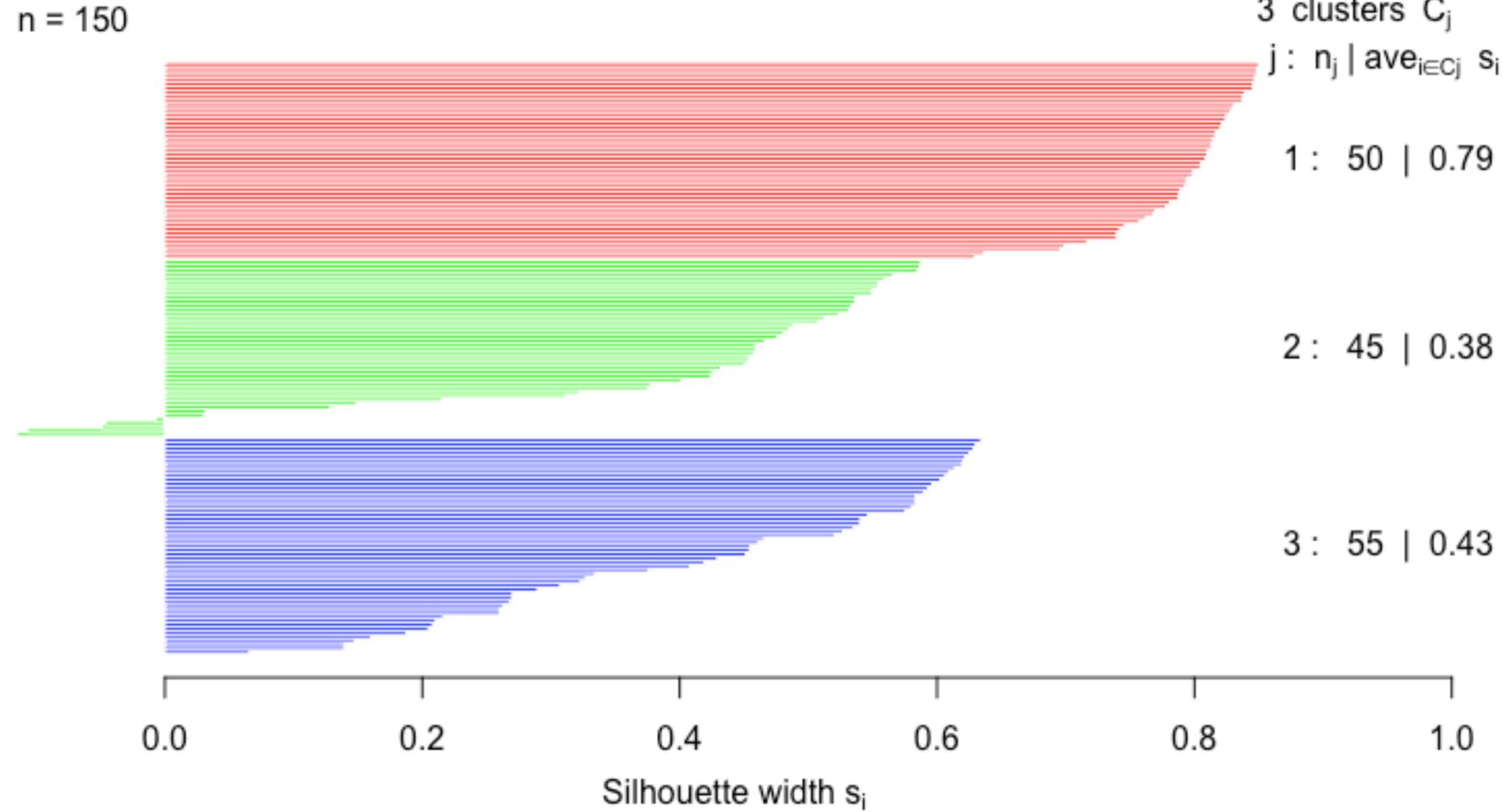
# Ejemplo Iris

- ▶ Elegir el grupo con la probabilidad más grande

<b>Observación</b>	<b>Grupo 1</b>	<b>Grupo 2</b>	<b>Grupo 3</b>
<b>102</b>	0.32	0.22	<b>0.45</b>
<b>143</b>	0.08	0.16	<b>0.75</b>
<b>57</b>	0.09	<b>0.65</b>	0.25
<b>71</b>	0.07	0.44	<b>0.47</b>
<b>139</b>	0.06	<b>0.72</b>	0.20
<b>84</b>	0.05	<b>0.69</b>	0.25
<b>114</b>	0.05	<b>0.75</b>	0.18
<b>122</b>	0.11	<b>0.61</b>	0.27
<b>73</b>	0.14	0.28	<b>0.56</b>
<b>52</b>	0.09	<b>0.63</b>	0.27
<b>55</b>	0.07	<b>0.65</b>	0.27
<b>66</b>	0.11	<b>0.62</b>	0.26
<b>76</b>	0.05	<b>0.69</b>	0.25

# Ejemplo Iris

## Silhouette



Average silhouette width : 0.54

Obs	Cluste	Alternativa	Silhouette
77	2	3	-0.049675159
124	2	3	-0.106637328
134	2	3	-0.046329606
147	2	3	-0.007183398
150	2	3	-0.114522411