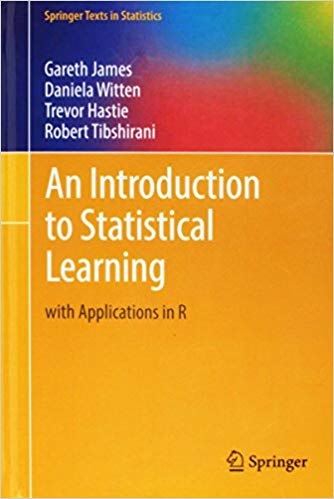
LAB10\_Apredizaje No Supervisado

#### We

#### 2 de marzo de 2020

(Tomado de Introduction to Statistical Learning with Applications in R, by G. James,D. Witten, T.Hastie, R. Tibshirani)



#install.packages("ISLR")  
library(ISLR)  
data("USArrests")  
states =row.names(USArrests )  
head(USArrests)

## Murder Assault UrbanPop Rape  
## Alabama 13.2 236 58 21.2  
## Alaska 10.0 263 48 44.5  
## Arizona 8.1 294 80 31.0  
## Arkansas 8.8 190 50 19.5  
## California 9.0 276 91 40.6  
## Colorado 7.9 204 78 38.7

# Análisis de componentes principales (PCA)

Primero examinamos brevemente los datos. Notamos que las variables poseen medias muy diferentes

apply(USArrests , 2, mean)

## Murder Assault UrbanPop Rape   
## 7.788 170.760 65.540 21.232

Vemos que hay en promedio tres veces más violaciones que asesinatos, y más de ocho veces más asaltos que violaciones. También podemos examinar las varianzas de las cuatro variables usando la función \(\texttt{apply()}\).

apply(USArrests , 2, var)

## Murder Assault UrbanPop Rape   
## 18.97047 6945.16571 209.51878 87.72916

No es sorprendente que las variables también tengan varianzas muy diferentes: La variable \(\texttt{UrbanPop}\) mide el porcentaje de la población en cada estado que vive en el área urbana, dicho número no es comparable al número de violaciones en cada estado por cada 100,000 individuos. Si no logramos escalar las variables antes de realizar PCA, entonces la mayoría de las componentes principales que observamos serían impulsadas por la variable Asalto, ya que tiene de lejos la mayor media y varianza. Por lo tanto, es importante estandarizar las variables para tener media cero y desviación estándar de uno, antes de realizar PCA.

Ahora realizamos el análisis de componentes principales utilizando la función \(\texttt{prcomp()}\), la cual es una de varias funciones en R que realizan PCA.

pr.out =prcomp (USArrests , scale =TRUE)

Por defecto, la función \(\texttt{prcomp()}\) centra las variables para que tengan una media cero. También al usar la opción \(\texttt{scale = TRUE}\), escalamos las variables para tener una desviación estándar de uno. La salida de \(\texttt{prcomp()}\) contiene varias cantidades útiles.

names(pr.out )

## [1] "sdev" "rotation" "center" "scale" "x"

Los componentes de \(\texttt{centro}\) y \(\texttt{escala}\) corresponden a los medias y a la desviación estándar de las variables que fueron escaladas antes de implementar PCA.

pr.out$center

## Murder Assault UrbanPop Rape   
## 7.788 170.760 65.540 21.232

pr.out$scale

## Murder Assault UrbanPop Rape   
## 4.355510 83.337661 14.474763 9.366385

La matriz de rotación proporciona los pesos de las componentes principales; cada columna de \(\texttt{pr.out\$rotation}\) contiene el vector de pesos de la correspondiente componente principal.

pr.out$rotation

## PC1 PC2 PC3 PC4  
## Murder -0.5358995 0.4181809 -0.3412327 0.64922780  
## Assault -0.5831836 0.1879856 -0.2681484 -0.74340748  
## UrbanPop -0.2781909 -0.8728062 -0.3780158 0.13387773  
## Rape -0.5434321 -0.1673186 0.8177779 0.08902432

Esta función se denomina matriz de rotación, porque cuando multiplicamos la matriz Matriz \(\boldsymbol{X}\) por \(\texttt{pr.out\$rotation}\), nos da las coordenadas de los datos originales en el sistema coordinado rotado. Estas coordenadas son los puntajes (scores) de las componentes principales.

Vemos que hay cuatro componentes principales distintos. Esto es esperado porque hay en general \(\min(n - 1, p)\) principal informativo componentes en un conjunto de datos con n observaciones y p variables.

Con el uso de la función \(\texttt{prcomp()}\), no necesitamos multiplicar explícitamente los datos por los vectores de pesos de las componentes principales para obtener los vectores de scores de componentes principales. Recuerde que \[\begin{equation} \boldsymbol{X}\Phi=Z. \end{equation}\]

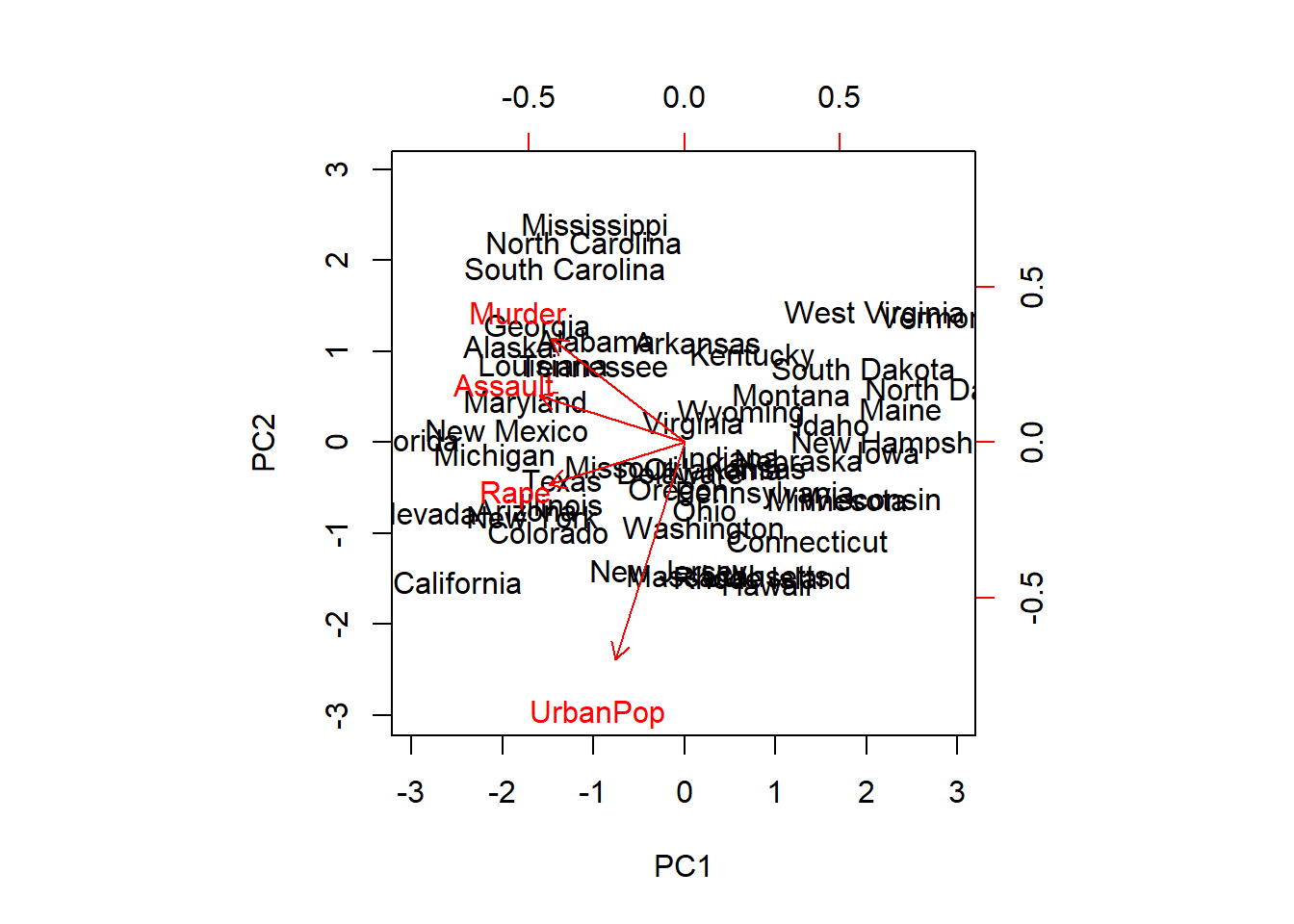
Más bien, la matriz \(50\times 4\) (\(\boldsymbol{Z}\)) posee como columnas los vectores de puntuación de las componentes principales. Es decir, la columna \(k\) de \(\boldsymbol{Z}\) corresponde al \(k\)-ésimo vector de scores de la componente principal correspondiente.

dim(pr.out$x)

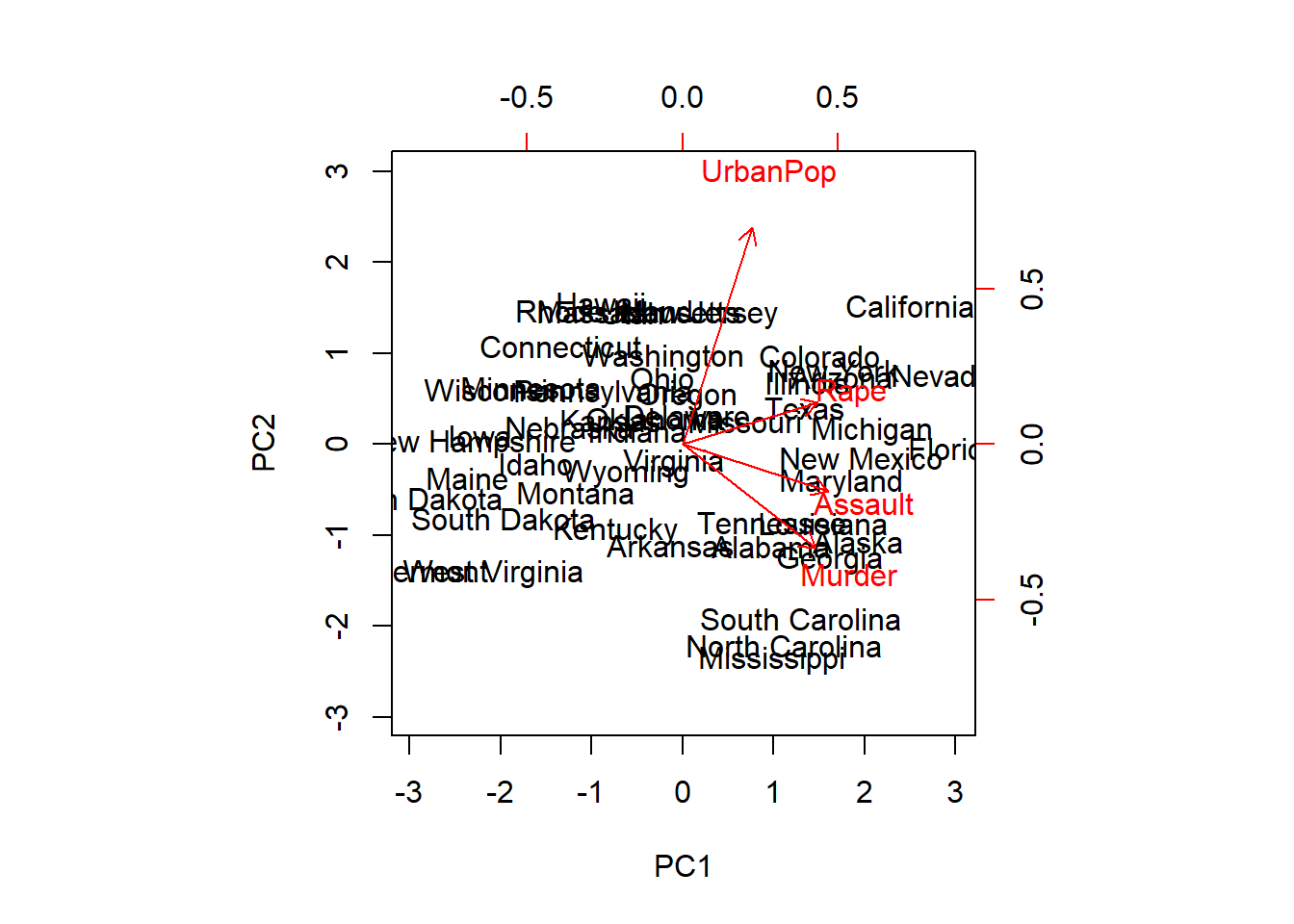
## [1] 50 4

Podemos realizar un plot de los dos primeros componentes principales de la siguiente manera

biplot(pr.out,scale =0)

 El argumento \(\texttt{scale = 0}\) para \(\texttt{biplot()}\) asegura que las flechas se escalen para representar los pesos; otros valores para la escala dan origen a biplots ligeramente diferentes y con diferentes interpretaciones. Observe que esta figura es una imagen especular (espejo) de la figura 10.1. Recordar que las componentes principales son únicas hasta un cambio de signo, por lo que se puede reproducir la figura 10.1 haciendo algunos pequeños cambios:

pr.out$rotation=-pr.out$rotation  
pr.out$x=-pr.out$x  
biplot (pr.out , scale =0)



La función \(prcomp()\) también proporciona la desviación estándar de cada componente principal. Por ejemplo, en el conjunto de datos de \(\texttt{USArrests}\), podemos acceder a estas desviaciones estándar:

pr.out$sdev

## [1] 1.5748783 0.9948694 0.5971291 0.4164494

Elevando estas al cuadrado se consigue la varianza explicada por cada componente

pr.var =pr.out$sdev ^2  
pr.var

## [1] 2.4802416 0.9897652 0.3565632 0.1734301

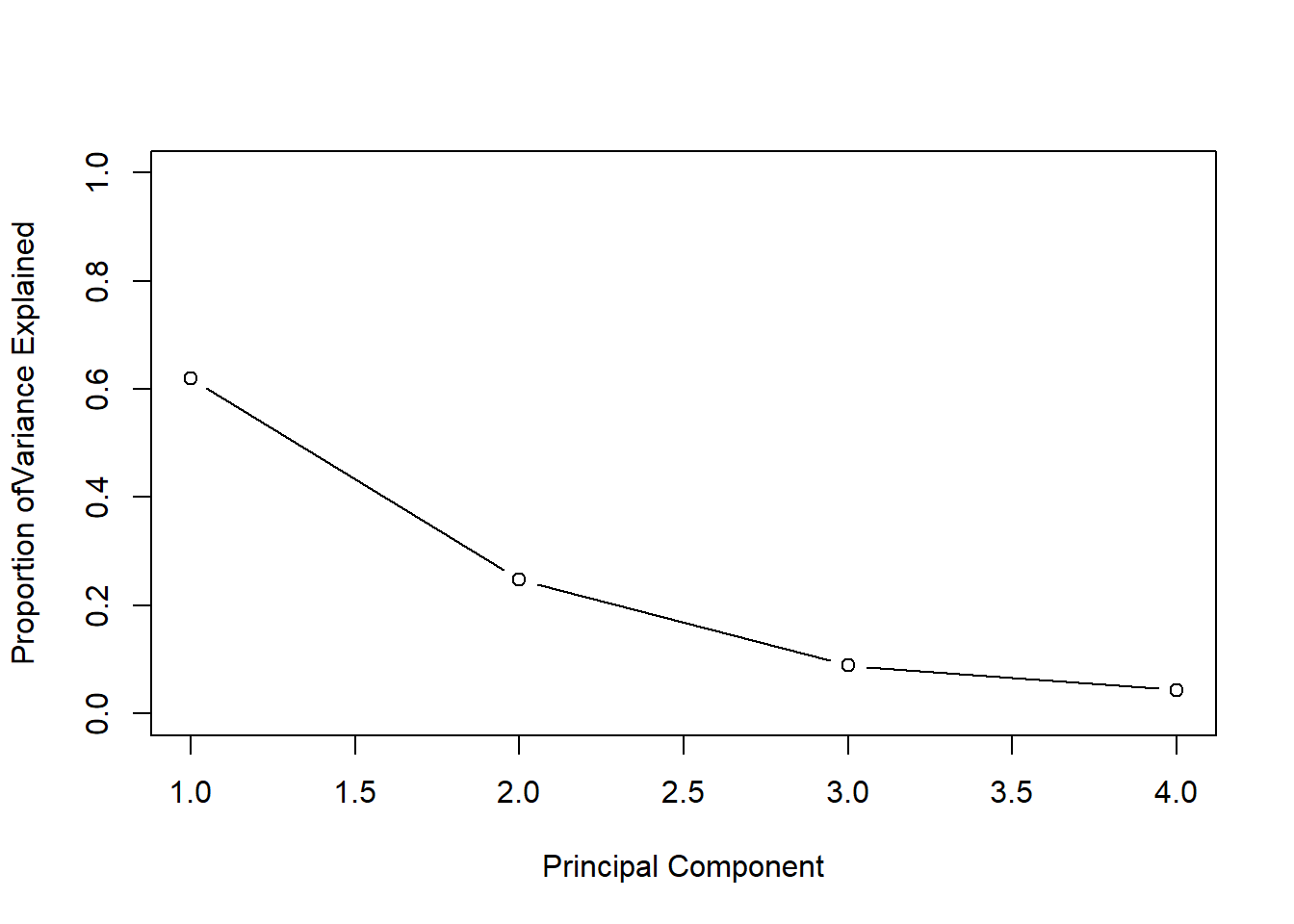
Para calcular la proporción de varianza explicada por cada componente principal, simplemente dividimos la varianza explicada por cada componente principal por la varianza total explicada por los cuatro componentes principales:

pve=pr.var/sum(pr.var )  
pve

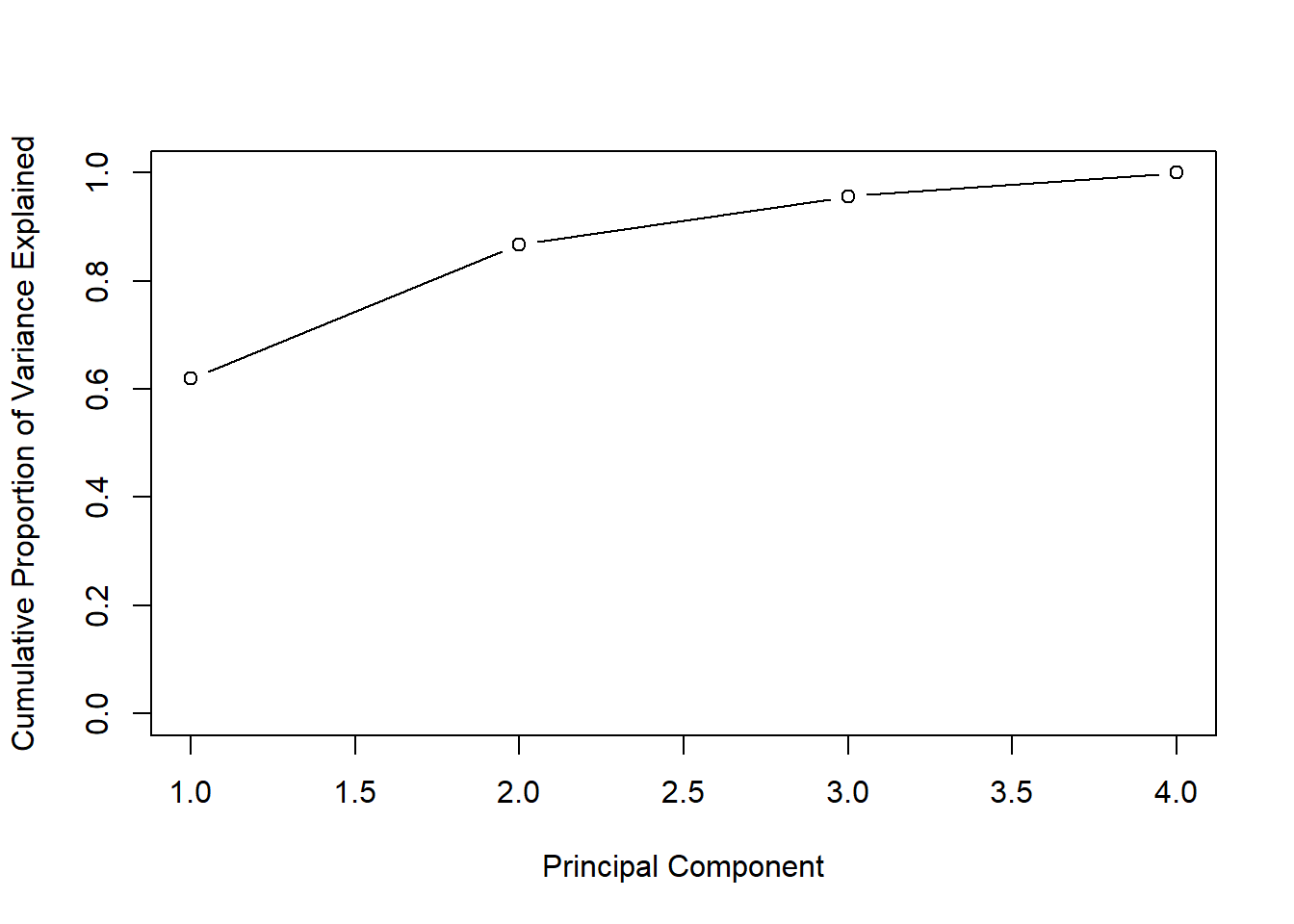
## [1] 0.62006039 0.24744129 0.08914080 0.04335752

se puede ver que el primer componente principal explica el 62.0% de la varianza en los datos, el siguiente componente principal explica el 24.7% de la varianza, Etcétera. También se puede realizar un plot de la PVE explicada por cada componente como la PVE acumulada, como sigue

plot(pve, xlab="Principal Component", ylab="Proportion ofVariance Explained",ylim=c(0,1) ,type='b')



plot(cumsum(pve), xlab="Principal Component", ylab ="Cumulative Proportion of Variance Explained", ylim=c(0,1),type='b')



# Clustering o agrupamiento por K medias

La función \(\texttt{kmeans()}\) realiza la agrupación de K-medias (k-means) en R. Un ejemplo simulado simple en el que realmente hay dos grupos en los datos: las primeras 25 observaciones tienen un cambio medio en relación con las siguientes 25 observaciones

set.seed (2)  
x=matrix (rnorm (50\*2) , ncol =2)  
x[1:25 ,1]=x[1:25 ,1]+3  
x[1:25 ,2]=x[1:25 ,2] -4

Ahora se lleva a cavo clustering k-medias con \(K=2\)

km.out =kmeans (x,2, nstart =20)

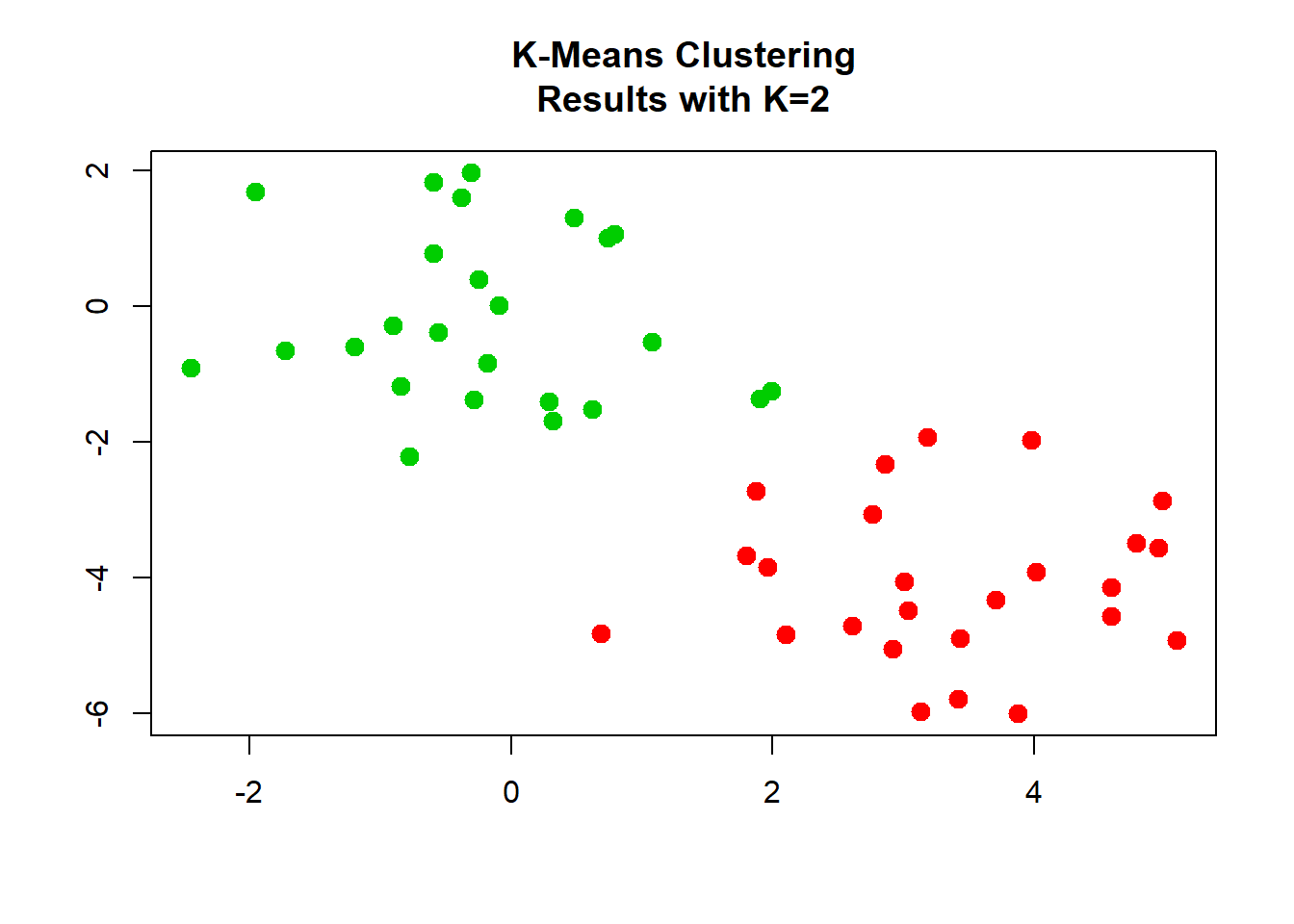
Las asignaciones de las 50 observaciones a los clusters están contenidas en \(\texttt{km.out\$cluster.}\)

km.out$cluster

## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [39] 2 2 2 2 2 2 2 2 2 2 2 2

El agrupamiento de K-medias separó perfectamente las observaciones en dos grupos aunque no proporcionamos ninguna información de grupo a \(\texttt{kmeans()}\). Se puede realizar un plot de los datos, con cada observación coloreada de acuerdo con su grupo asignación

plot(x, col =(km.out$cluster +1) , main="K-Means Clustering  
Results with K=2", xlab ="", ylab="", pch =20, cex =2)



Aquí las observaciones pueden trazarse fácilmente porque son bidimensionales. Si hubiera más de dos variables, podríamos realizar PCA y graficar los dos primeros vectores de scores de las componentes principales.

En este ejemplo, sabíamos que realmente había dos grupos porque Generamos los datos. Sin embargo, para datos reales, en general no sabemos El verdadero número de grupos. En cambio, podríamos haber realizado K-means agrupamiento en este ejemplo con \(K = 3\).

set.seed (4)  
km.out =kmeans (x,3, nstart =20)  
km.out

## K-means clustering with 3 clusters of sizes 17, 23, 10  
##   
## Cluster means:  
## [,1] [,2]  
## 1 3.7789567 -4.56200798  
## 2 -0.3820397 -0.08740753  
## 3 2.3001545 -2.69622023  
##   
## Clustering vector:  
## [1] 1 3 1 3 1 1 1 3 1 3 1 3 1 3 1 3 1 1 1 1 1 3 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [39] 2 2 2 2 2 3 2 3 2 2 2 2  
##   
## Within cluster sum of squares by cluster:  
## [1] 25.74089 52.67700 19.56137  
## (between\_SS / total\_SS = 79.3 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

Cuando \(K = 3\), K-medias divide los dos grupos “verdaderos”. Para ejecutar la función \(\texttt{kmeans()}\) en R con múltiples asignaciones iniciales de clúster, se utiliza el argumento \(\texttt{nstart}\). Si un valor de \(\texttt{nstart}\) mayor que uno se utiliza, entonces el agrupamiento K-medias se realizará utilizando múltiples asignaciones aleatorias en el Paso 1 del Algoritmo 10.1, y la función \(\texttt{kmeans()}\) informa solo los mejores resultados. Aquí comparamos usando \(\texttt{nstart = 1}\) con \(\texttt{nstart = 20}\).

set.seed (3)  
km.out =kmeans (x,3, nstart =1)  
km.out$tot.withinss

## [1] 97.97927

km.out =kmeans (x,3, nstart =20)  
km.out$tot.withinss

## [1] 97.97927

Tener en cuenta que \(\texttt{km.out\$tot.withinss}\) es la suma total de cuadrados intra- clúster,que buscamos minimizar mediante la agrupación de K-means (Ecuación 10.11). La suma de cuadrados individuales intra-clúster está contenida en el vector \(\texttt{km.out\$withinss}\). Se recomienda fuertemente ejecutar siempre clustering K-medias con un valor de \(\texttt{nstart}\), como 20 o 50, ya que de lo contrario se puede optener un mínimo local no deseado.

Al realizar la agrupación de K-medias, además de usar múltiples asignaciones iniciales de clúster, también es importante establecer una semilla aleatoria utilizando la función \(\texttt{set.seed()}\). De esta manera, las asignaciones iniciales del clúster en el Paso 1 pueden replicarse, y la salida de K-medias es completamente reproducible.

# Clustering Jerarquico

La función \(\texttt{hclust()}\) implementa la agrupación jerárquica en R. En el siguiente ejemplo utilizamos los mimos datos de con que utilizamos k-medias (sección anterior). Para graficar el dendrograma de agrupamiento jerarquico utilizando enlace completo, única (single) y promedio, con la distancia euclidia como medida de disimilitud. Comenzamos por la agrupación de observaciones utilizando un enlace completo. Se utiliza la función \(\texttt{dist()}\) para calcular la matriz de distancia euclidia inter-observación 50 × 50.

hc.complete =hclust(dist(x), method ="complete")

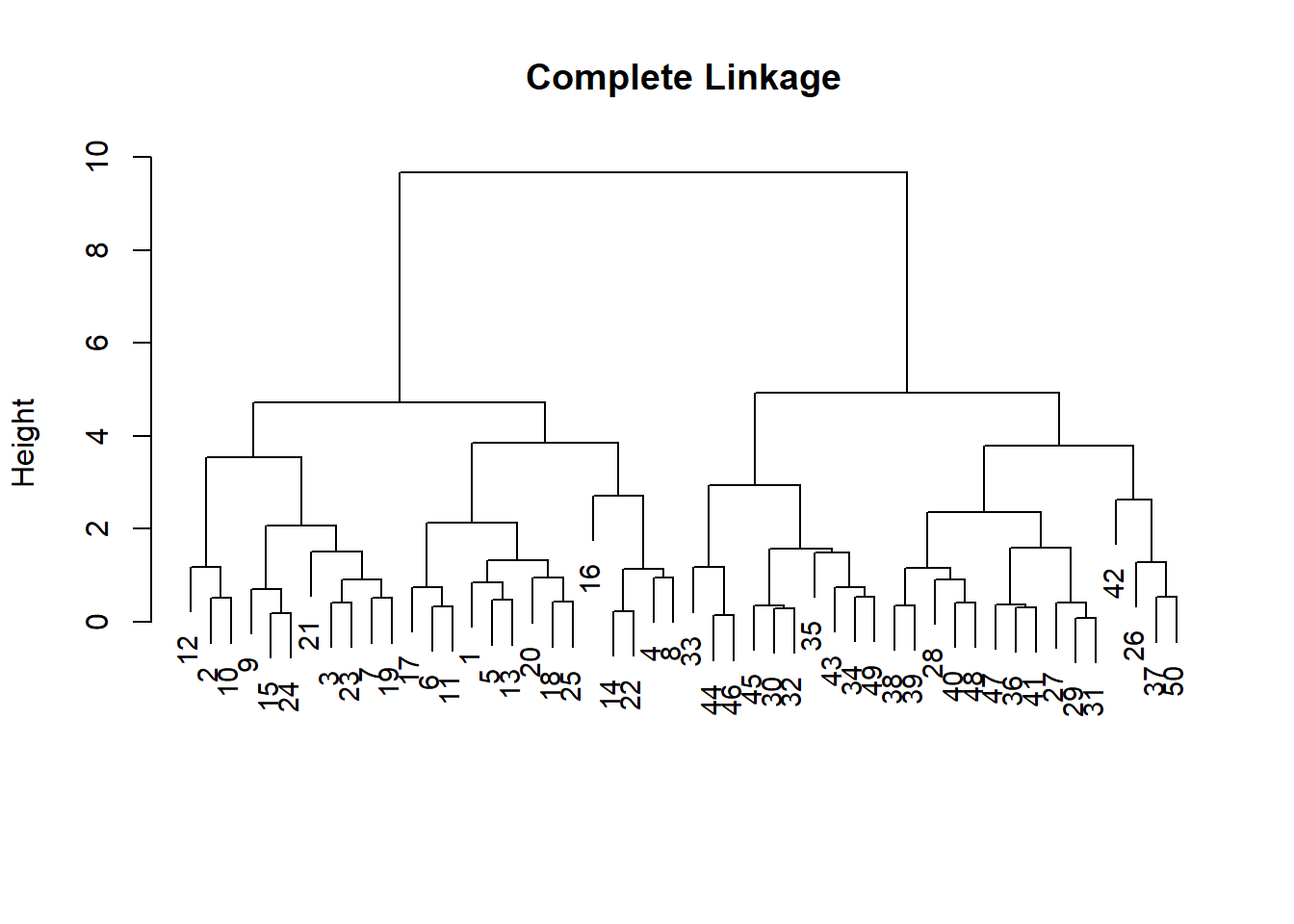
también se puede realizr un clusterin jerarquico con enlace single y promedio en vez

hc.average =hclust(dist(x), method ="average")

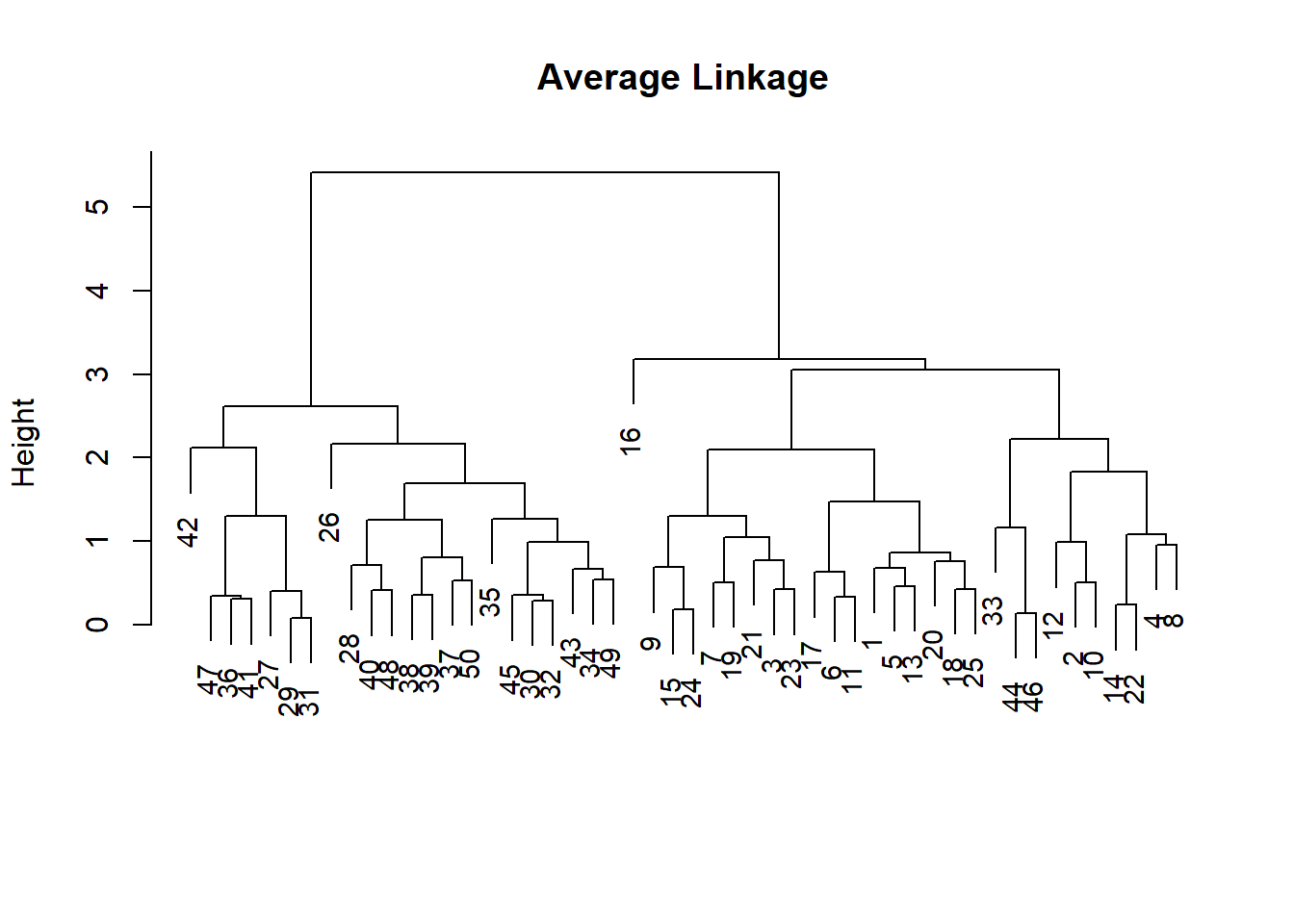
hc.single =hclust(dist(x), method ="single")

Ahora podemos hacer un plot los dendrogramas obtenidos usando la función usual \(\texttt{plot()}\). Los números en la parte inferior de la gráfica identifican cada observación.

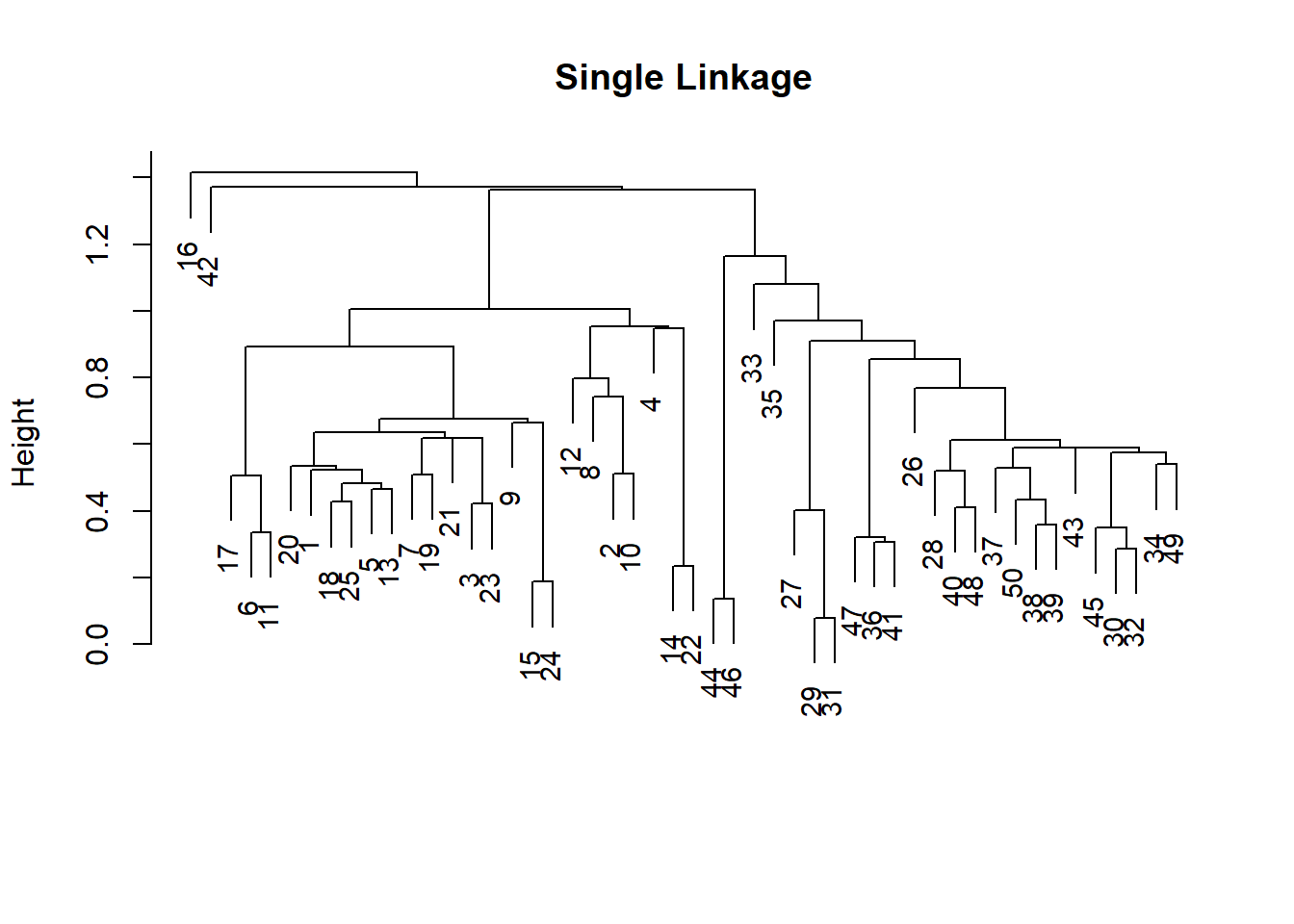
plot(hc.complete ,main =" Complete Linkage ", xlab="", sub ="",  
cex =.9)



plot(hc.average , main =" Average Linkage ", xlab="", sub ="",  
cex =.9)



plot(hc.single , main=" Single Linkage ", xlab="", sub ="",  
cex =.9)



Para determinar las etiquetas de clúster para cada observación asociada con un corte del dendrograma, se puede usar la función \(\texttt{cutree()}\):

cutree (hc.complete, 2)

## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2  
## [39] 2 2 2 2 2 2 2 2 2 2 2 2

cutree (hc.average, 2)

## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2  
## [39] 2 2 2 2 2 1 2 1 2 2 2 2

cutree (hc.single,2)

## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [39] 1 1 1 1 1 1 1 1 1 1 1 1

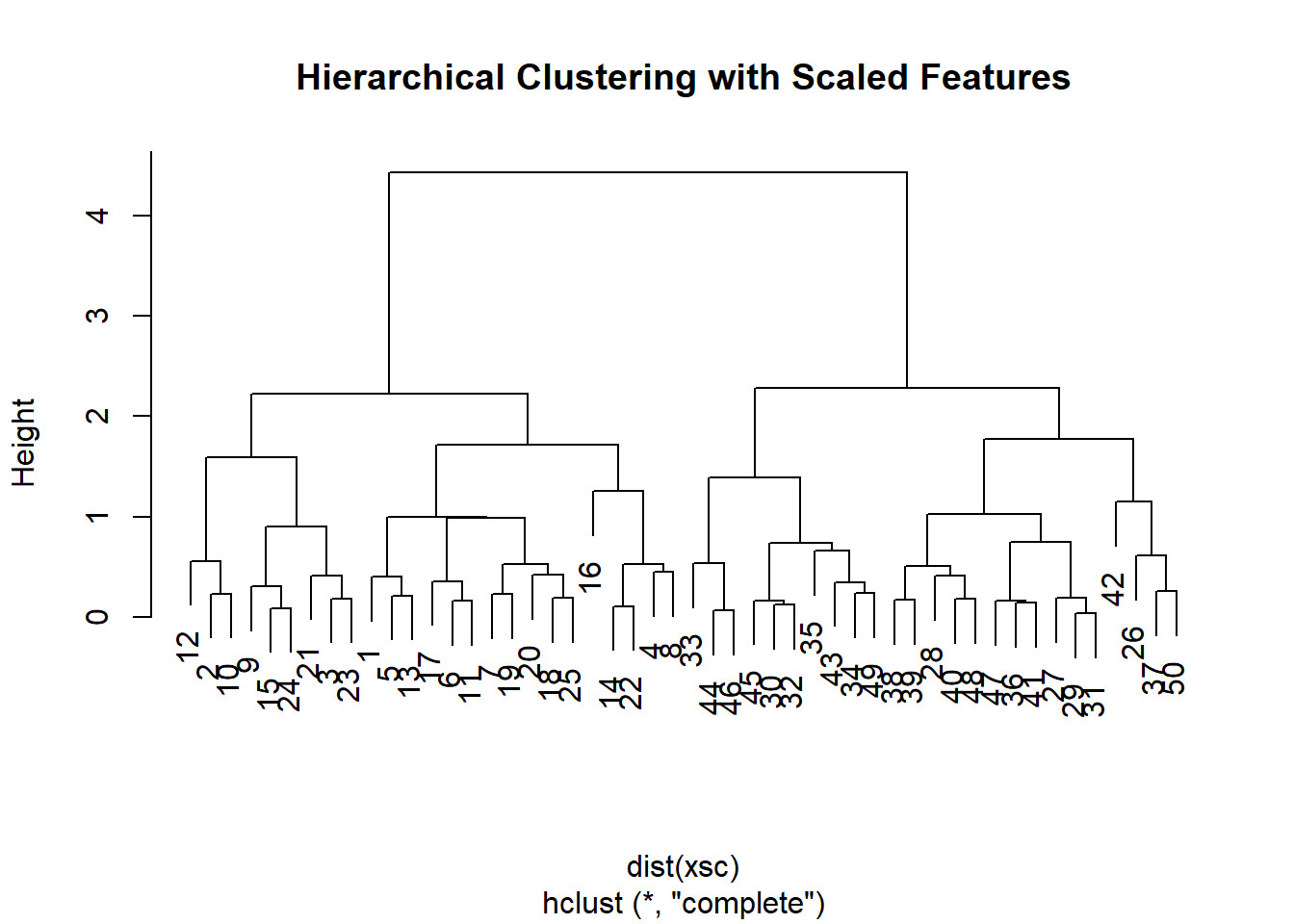
Para estos datos, el enlace completo y promedio generalmente separa las observaciones en sus grupos correctos. Sin embargo, el enlace único (single) identifica un punto como perteneciente a su propio grupo. Se obtiene una respuesta más sensata cuando Se seleccionan cuatro grupos, aunque todavía hay dos clusters de 1 observación (singleton).

cutree (hc.single , 4)

## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3  
## [39] 3 3 3 4 3 3 3 3 3 3 3 3

Para escalar las variables antes de realizar la agrupación jerárquica de observaciones, utilizamos la función \(\texttt{scale()}\):

xsc=scale(x)  
plot(hclust(dist(xsc), method ="complete"), main ="Hierarchical Clustering with Scaled Features")



La distancia basada en la correlación se puede calcular utilizando la función \(\texttt{as.dist()}\), que convierte una matriz simétrica cuadrada arbitraria en una forma que la función \(\texttt{hclust()}\) reconoce como una matriz de distancia. Sin embargo, esto solo tiene sentido para los datos con al menos tres características ya que la correlación absoluta entre dos observaciones cualquiera con mediciones en dos características es siempre 1. Por lo tanto, agruparemos un conjunto de datos tridimensionales.

x=matrix (rnorm (30\*3),ncol =3)  
dd=as.dist(1- cor(t(x)))  
plot(hclust(dd, method ="complete"), main="Complete Linkage  
with Correlation-Based Distance", xlab="", sub ="")

