K-MEANS

Lino Oswaldo Sanchez

27/5/2022

Introducción

Se parte de un amuestra de n elementos con p variable. donde el objetivo es dividir la muestra en un número de grupos prefijado,(k).Este algoritmo esta compuesto por cuatro etapas:

1.- Selecciona K putos como centro de los grupos iniciales

1.1.- Asignando aleatoriamente los objetos a los grupos y tomando los centros de los grupos formados.

1.2.- Tomando como centro los k puntos más lejanos.

1.3.- Construyendo los grupos con información a priori o seleccionando los centros (a priori).

2.- Calcular las distancias euclídeas de cada elemento al centro de los k grupos y asignar cada elemento al grupo próximo. La asignación se realiza secuencialmente y al introducir un nuevo elemento en un grupo se recalculan las coordenadas de la nueva media de grupo.

3.- Definir un criterio de optimalidad y comprobar si reasignando uno a uno cada elemento de un grupo a otro mejora el criterio.

4.- Sí no es posible mejorar el criterio de optimalidad, terminar el proceso.

Matriz de datos.

```
X<-as.data.frame(state.x77)
colnames(X)
```

```
## [1] "Population" "Income"     "Illiteracy" "Life Exp"   "Murder"
## [6] "HS Grad"    "Frost"      "Area"
```

Usamos una matriz de datos precargada en r llamada **state.x77** y con (*colnames*) observamos el nombre de las variables y recordando de ejercicios anteriores tres variables, tendrán que ser modificadas pero eso sera más adelanta.

Transformación de las variables x1,x3 y x8

Con la función de logaritmo como ya mencionamos se modificaran estas tres variables, porqué las cantidades dentro de ellas son muy elevadas y sacándoles el logaritmo las cantidades serán mas pequeñas. Esto se hace para que cuando estemos usando el algoritmo nuestros cálculos salgan mejor, también cabe recalcar que se renombran esas variables sumando les el prefijo “log” y su nombre original.

```r
X[,1]<-log(X[,1])
colnames(X)[1]<-"Log-Population"

X[,3]<-log(X[,3])
colnames(X)[3]<-"Log-Illiteracy"

X[,8]<-log(X[,8])
colnames(X)[8]<-"Log-Area"
```

Separación de filas y columnas.

Como ya se atrabajado esta base de datos sabemos la dimención, pero aún así vemos que esta conformada por 50 observaciones y 8 variables

```r
dim(X)
```

```
## [1] 50  8
```

Filas

```r
n<-dim(X)[1]
```

Columnas

```r
p<-dim(X)[2]
```

Estandarizacion univariante.

Ya que las variables no tienen la misma unidad e medida estandarizamos escalando y creando una matriz.

```r
X.s<-scale(X)
```

Algoritmo k-medias 3 grupos

nstart: cantidad de subconjuntos aleatorios que se escogen para realizar los cálculos de algoritmo. X.S: estadarización de los datos donde usaremos tres grupos

```r
Kmeans.3<-kmeans(X.s, 3, nstart=25)
```

centroides

Extraemos los centroides de el objeto Kmeans.3 estos son los centroides del que se usarna para los clusters

```r
Kmeans.3$centers
```

```
##   Log-Population     Income Log-Illiteracy   Life Exp      Murder    HS Grad
## 1     -0.7900149  0.2080926    -0.93960948  0.5642988 -0.71791785  0.7707484
## 2      0.2360549 -1.2266128     1.31921387 -1.0778757  1.10983501 -1.3566922
## 3      0.5693805  0.5486843     0.05412021  0.1388564 -0.01977495  0.1203417
##        Frost   Log-Area
## 1  0.8803670  0.4093602
## 2 -0.7719510  0.1991243
## 3 -0.3291597 -0.4878988
```

cluster de pertenencia

Extraemos y visualizamos en que cluster esta las observaciones.

```
Kmeans.3$cluster
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              2              1              3              2              3
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              1              3              3              3              2
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              1              3              3              1
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              1              2              2              1              3
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              3              3              1              2              3
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              1              1              1              1              3
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              2              3              2              1              3
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              3              1              3              3              2
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              1              2              2              1              1
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              3              3              2              1              1
```

Suma de cuadrados dentro de los grupos SCDG

El criterio de homogeneidad que se utiliza en el algoritmo de k-medias es la suma de cuadrados dentro de los grupos (SCDG) para todas las variables.

Esto equivale a la suma ponderada de las varianzas de las variables en los grupos; lo que se busca en concreto es que la suma de cuadrados se ala menor posibles puesto que entre mas pequeña se la varianza los grupos serán más homogéneas.

```
SCDG<-sum(Kmeans.3$withinss)
SCDG
```

```
## [1] 203.2068
```

5.- separa los Clusters

Separamos los clusters y los ponemos creamos un objeto.

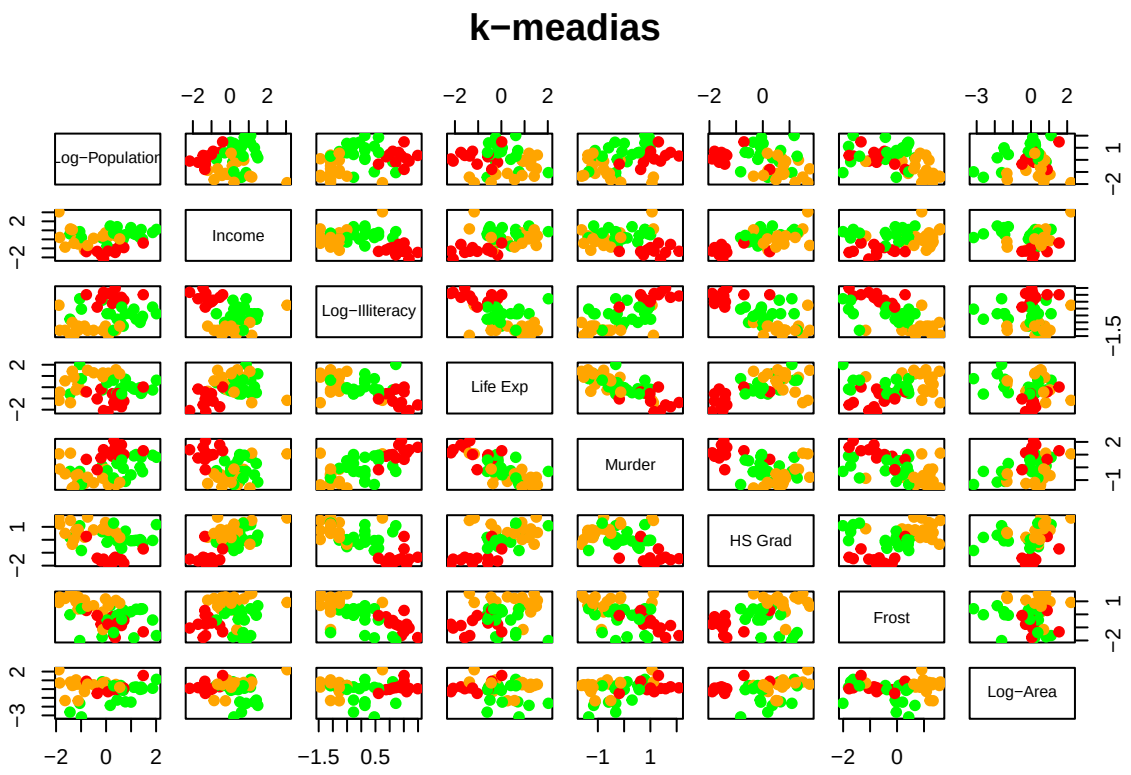
```
cl.kmeans<-Kmeans.3$cluster
```

De ser necesario lo podríamos visualizar y aseguramos que este lo que queremos pero podemos no hacerlo para ahorrar espacio en este documento.

Scatter plot con la division de grupos

Obtenidos (se utiliza la matriz de datos centrados).

```
col.cluster<-c("orange", "red", "green")[cl.kmeans]
pairs(X.s, col=col.cluster, main="k-meadias", pch=19)
```



Lo que podemos visualizar aquí es como están agrupados las observaciones y su correlación entre ellas, como se puede ver claramente las variables que se encuentran en la parte del centro son las que tiene una mejor correlación a comparación de las observadas en el exterior del gráfico.

Visualizacion con las dos componentes principales

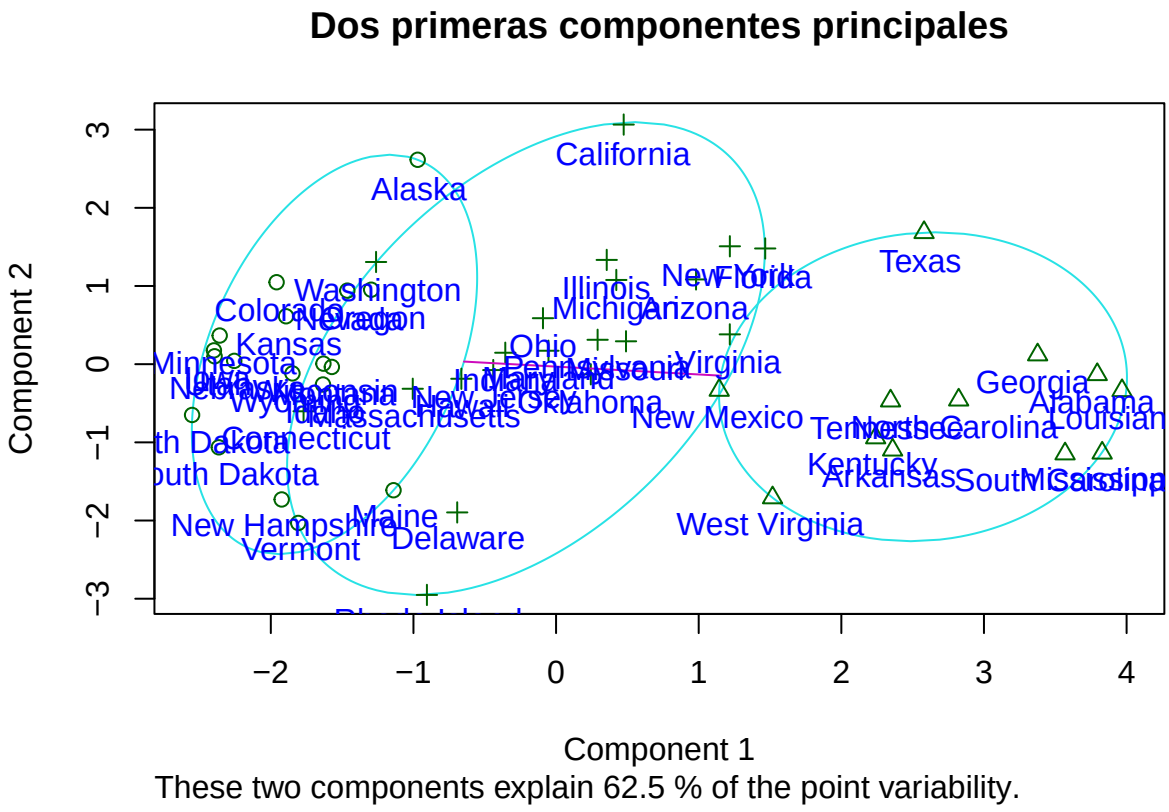
Con funciones que se usan en componentes principales para poder ver la visualización de los cluster.

librería necesaria

```
library(cluster)
```

```
clusplot(X.s, cl.kmeans,
         main="Dos primeras componentes principales")

text(princomp(X.s)$score[,1:2],
     labels=rownames(X.s), pos=1, col="blue")
```



Podemos ver la agrupación de los clusters y saber cual clusters es cada uno con ayuda de la lista de agrupamiento y la figura de cada agrupación podemos decir que el grupo en el que esta Texas es el cluster 2, California cluster 1 y Alaska cluster 3.

También nos indica que en estos dos componentes principales se explica el 62.5% de la variabilidad podemos decir que es buena.

Silhouette

Representación gráfica de la eficacia de clasificación de una observación dentro de un grupo.

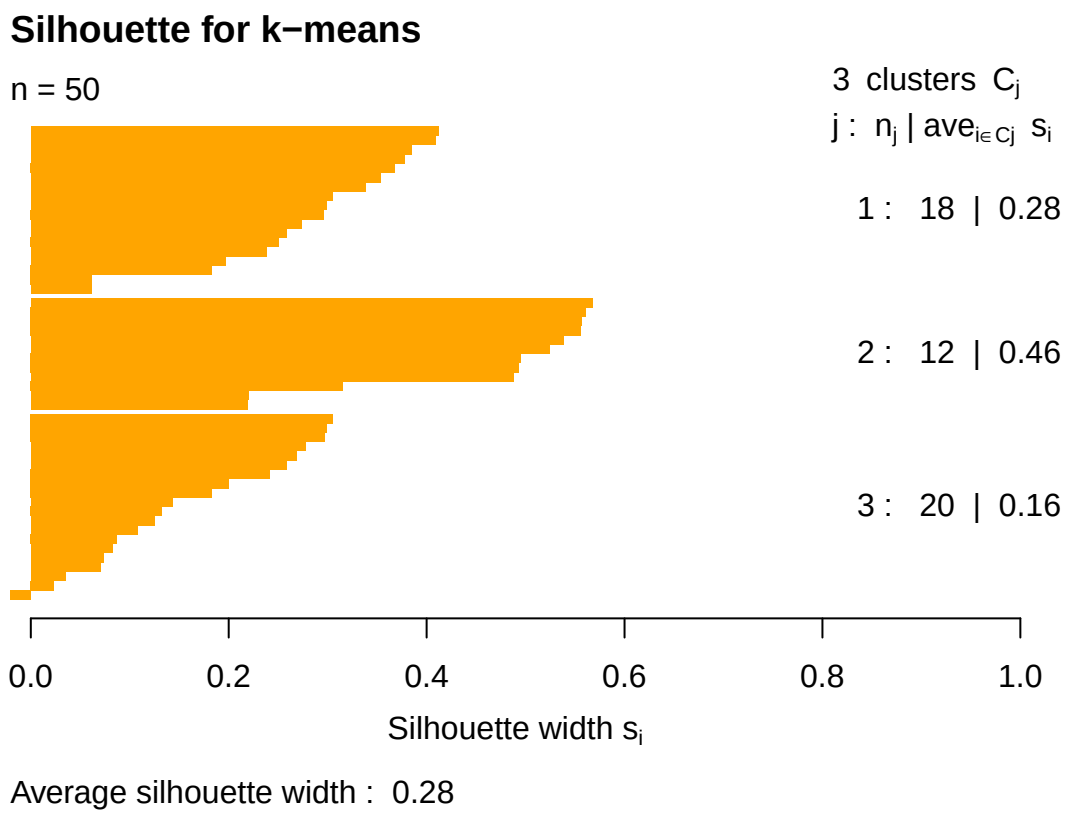
Generacion de los calculos

Para realizar estos cálculos nos apoyamos de la distancia euclidia, de la matriz escalada y creamos un objeto para realizar el silhouette con la distancia euclidia calculada y los clusters.

```
dist.Euc<-dist(X.s, method = "euclidean")
Sil.kmeans<-silhouette(cl.kmeans, dist.Euc)
```

Generacion del grafico

```
plot(Sil.kmeans, main="Silhouette for k-means",
col="orange")
```



Interpretación

Se puede ver que el anchopara cada silhouette por cluster es medianamente bueno en el 1 y 2 y en el 3 es mas abajo a comparacion además de presentar una obsevción negativa, esto puede ser por que en el agrupamiento alguanas obsrvaciones se traslapan en los grupos.

Adicional el **average silhouette width**(ancho medio del silhouette)=0.28 no es muy alto debería ser mejor par estar seguros de que la cantidad de cluster que proponemos es la correcta.

Ejercicio

1. replicar el script pero con un numero de clusters diferentes a 3 y 1

2. incluir la interpretación del silhouetter

Para este ejercicio el describir cada paso seria redundante pues ya esta descrito en a primera parte de este reporte, así que sera lo mismo la parte que cambiara es la cantidad de clusters que se harán solo eso.

Cargar la matriz de datos.

```
x<-as.data.frame(state.x77)
colnames(x)
```

```
## [1] "Population" "Income"     "Illiteracy" "Life Exp"   "Murder"
## [6] "HS Grad"    "Frost"      "Area"
```

Transformacion de las variables x1,x3 y x8 con la funcion de logaritmo.

```
x[,1]<-log(x[,1])
colnames(x)[1]<-"Log-Population"

x[,3]<-log(x[,3])
colnames(x)[3]<-"Log-Illiteracy"

x[,8]<-log(x[,8])
colnames(x)[8]<-"Log-Area"
```

Algoritmo k-means

Separacion de filas y columnas.

```
dim(x)
```

```
## [1] 50  8
```

filas

```
n<-dim(x)[1]
```

columnas

```
p<-dim(x)[2]
```

Estandarizacion univariante.

```
x.s<-scale(x)
```

Algoritmo k-medias con 8 grupos

nstart: cantidad de subconjuntos aleatorios que se escogen para realizar los cálculos de algoritmo. X.S: estadarización de los datos

```
Kmeans.3<-kmeans(x.s, 8, nstart=25)
```

centroides

```
Kmeans.3$centers
```

```
##   Log-Population     Income Log-Illiteracy    Life Exp     Murder     HS Grad
## 1     0.12233125 -1.3014617     1.30192615 -1.17731360  1.0919809 -1.41578257
## 2     1.02429084  0.2250901    -0.11672027 -0.18767472  0.3520964 -0.21146471
## 3    -1.05349612  0.8579753     1.21713430  2.02727434 -0.3191080  1.08852326
## 4     1.11446170  0.3677153     0.80175375  0.05691327  0.7779952  0.25591192
## 5    -1.65470747  2.1094604    -0.34909739 -1.27280111  1.0895183  1.58994719
## 6    -1.30355300 -0.2681986    -0.97758128  0.35488847 -0.9218376  0.46019574
## 7    -0.02012796  0.2632441    -1.05275367  1.16562936 -0.9511840  0.92206977
## 8    -0.00827024  0.9198172     0.04979557  0.26673748 -0.7028684  0.03408857
##         Frost    Log-Area
## 1 -0.72065003  0.07602772
## 2  0.08733986  0.13224248
## 3 -2.00958630 -1.62183033
## 4 -1.62001974  0.92551369
## 5  1.26084899  1.51085951
## 6  1.15263606  0.03872450
## 7  0.30109380  0.49075236
## 8  0.19635437 -1.97583298
```

cluster de pertenencia

```
Kmeans.3$cluster
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              5              4              1              4
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              7              8              8              4              1
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              6              2              2              7
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              7              1              1              6              8
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              8              2              7              1              2
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              6              7              5              6              8
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              1              2              1              6              2
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
```

```
##              2              7              2              8              1
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              6              1              4              7              6
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              2              7              1              7              6
```

suma de cuadrados dentro de los grupos SCDG

```
SCDG<-sum(Kmeans.3$withinss)
SCDG
```

```
## [1] 97.45184
```

Con 8 grupos (clusters) la suma de cuadrados no mejora es to ya lo he hecho con más o menos cantidad de clusters y el resultado no mejora; esas simulaciones no están incluidas en este documento pues sería en demasía extenso.

Separa los Clusters

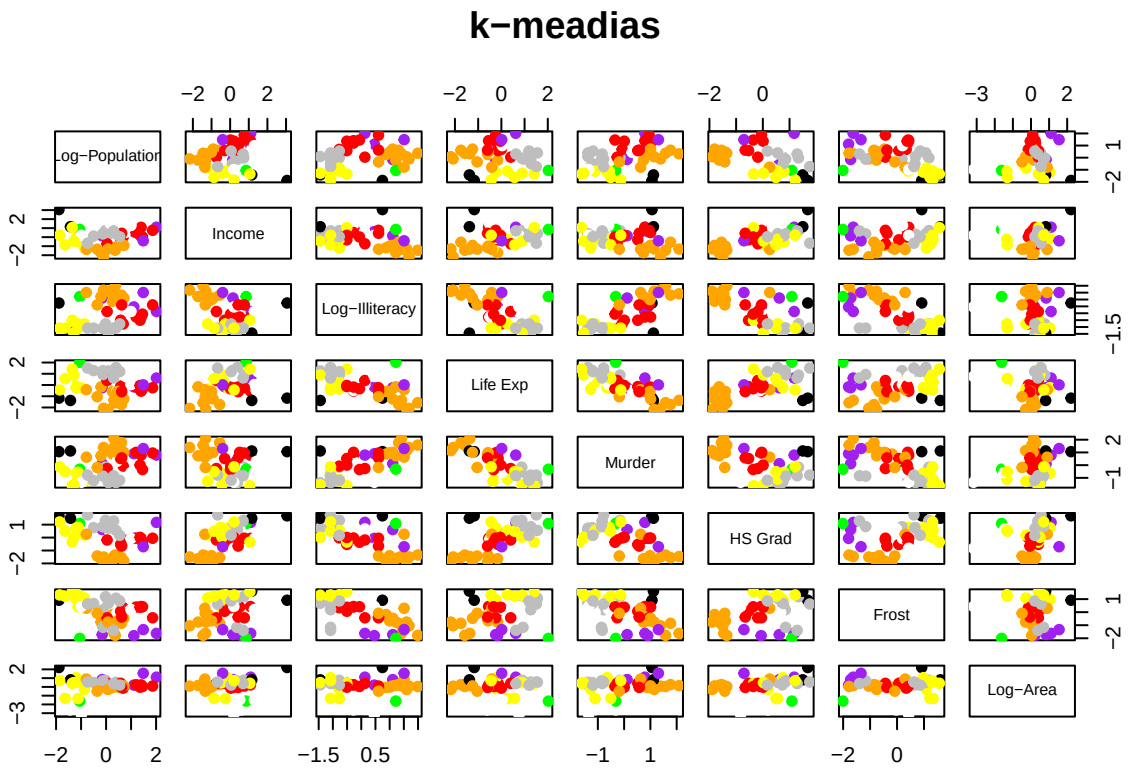
```
cl.kmeans<-Kmeans.3$cluster
cl.kmeans
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              5              4              1              4
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              7              8              8              4              1
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              6              2              2              7
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              7              1              1              6              8
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              8              2              7              1              2
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              6              7              5              6              8
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              1              2              1              6              2
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              2              7              2              8              1
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              6              1              4              7              6
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              2              7              1              7              6
```

Scatter plot con la division de grupos

obtenidos se utiliza la matriz de datos centrados.

```
col.cluster<-c("orange", "red", "green","purple","black","yellow","grey","white")[cl.kmeans]
pairs(x.s, col=col.cluster, main="k-meadias", pch=19)
```

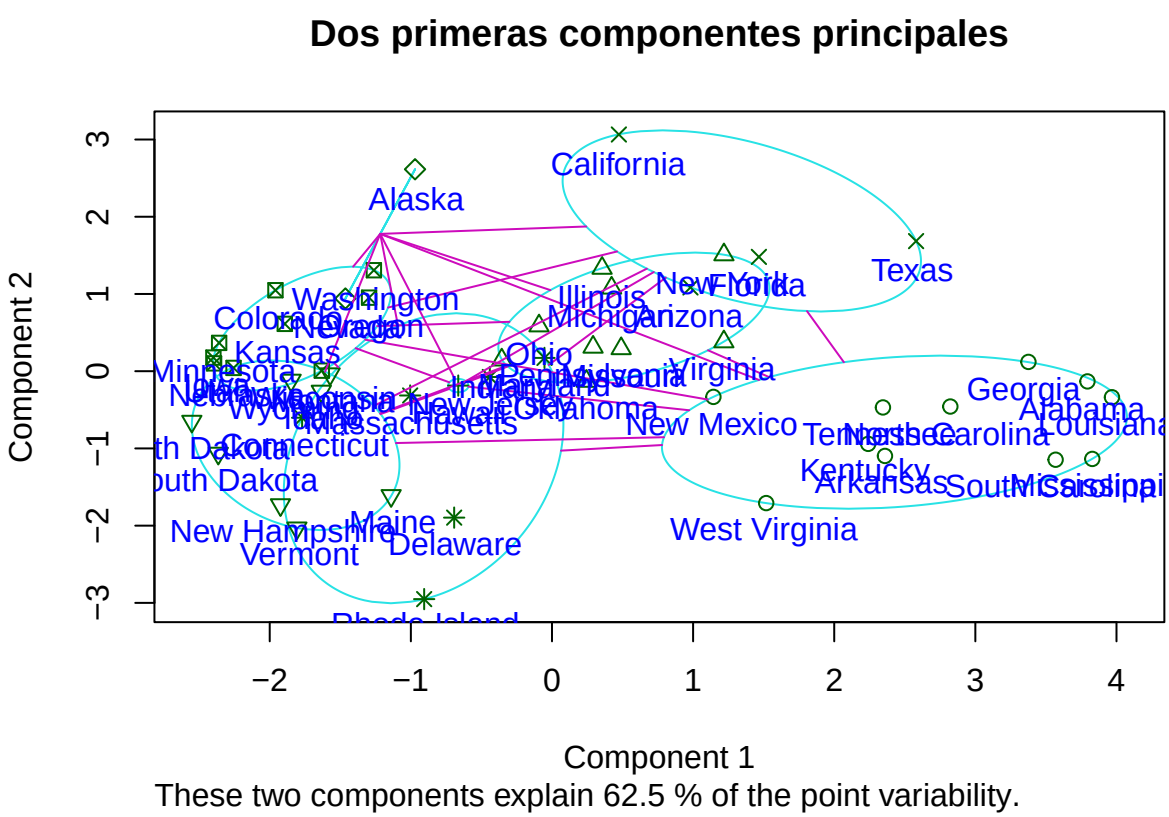


Visualizacion con las dos componentes principales

```
library(cluster)

clusplot(x.s, cl.kmeans,
         main="Dos primeras componentes principales")

text(princomp(x.s)$score[,1:2],
     labels=rownames(x.s), pos=1, col="blue")
```



These two components explain 62.5 % of the point variability.

Silhouette

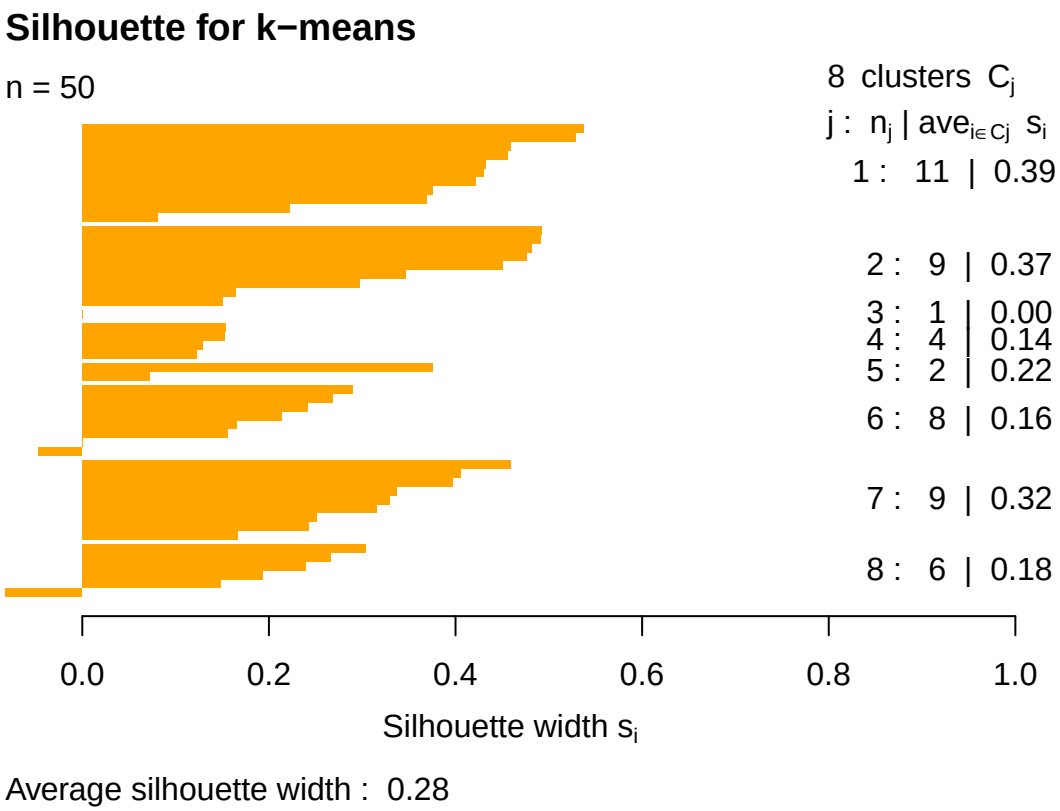
Representación gráfica de la eficacia de clasificación de una observación dentro de un grupo.

##Generacion de los calculos

```
dist.Euc<-dist(x.s, method = "euclidean")
Sil.kmeans<-silhouette(cl.kmeans, dist.Euc)
```

Generacion del grafico

```
plot(Sil.kmeans, main="Silhouette for k-means",
     col="orange")
```



Interpretación

Entre mas clutser se decida hacer tenemos en el Silhouette que en algunos clusters el ancho del Silhouette mejora en unos y empeora en otros teniendo en cuenta que entre mas cercano a uno este el ancho del Silhouette la decisión sería hacer solo dos clusters además la suma de cuadrados que da prácticamente igual incluso aumenta la suma por una décimas.