Algoritmo PAM

Lino Oswaldo Sanchez

29/5/2022

Introducción

- En este algoritmo se consideran las medianas.
- Modificación del algoritmo k-medias.
- Busca “k objetos representativos”.
- Es un método robusto para datos atípicos.

A pesar de su buen funcionamiento con conjunto de datos pequeños, no es lo suficientemente eficaz para agrupar numerosos conjuntos de datos.

Pasos del algoritmo

1.- Sea X i para i = 1, . . . , n el conjunto de observaciones de la matriz de datos.

2.- Calcula una matriz que contiene las distancias entre las n observaciones.

3.- Elige k observaciones como los “medoides” de los k grupos iniciales.

4.- Asigna a cada observación a su “medoide” más cercano usando la matriz de distancias D.

5.- Para cada cluster, se busca la observación X j (sí existe) que proporcioa la mayor reducción de la suma de cuadrados.

Algoritmo PARTITION AROUND MEDOIDS (PAM)

para las medianas a diferencia de k means que son medias #

```
library(cluster)
```

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"
```

Cargamos la matriz precargada en r y visualizamos su nombre para recodar que variables están dentro. # Transformacion de datos

```
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"
```

Como en ejercicio anteriores esta variables serán transformadas par que los análisis salgan lo mejor posible pues las cantidades contenidas dentro de las variables son muy grandes sacándoles el algoritmo podemos trabajar de mejor forma con ellas.

Algoritmo PAM

Separacion 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

```
dim(X)
```

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

filas

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

columnas

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

Estandarizacion univariante.

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

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

3.- Aplicacion del algoritmo

Aquí aplicamos el algoritmo PAM con tres grupos como en el de K-means a la matriz de las variables escaladas.

```
pam.3<-pam(X.s,3)
```

4.- Clusters

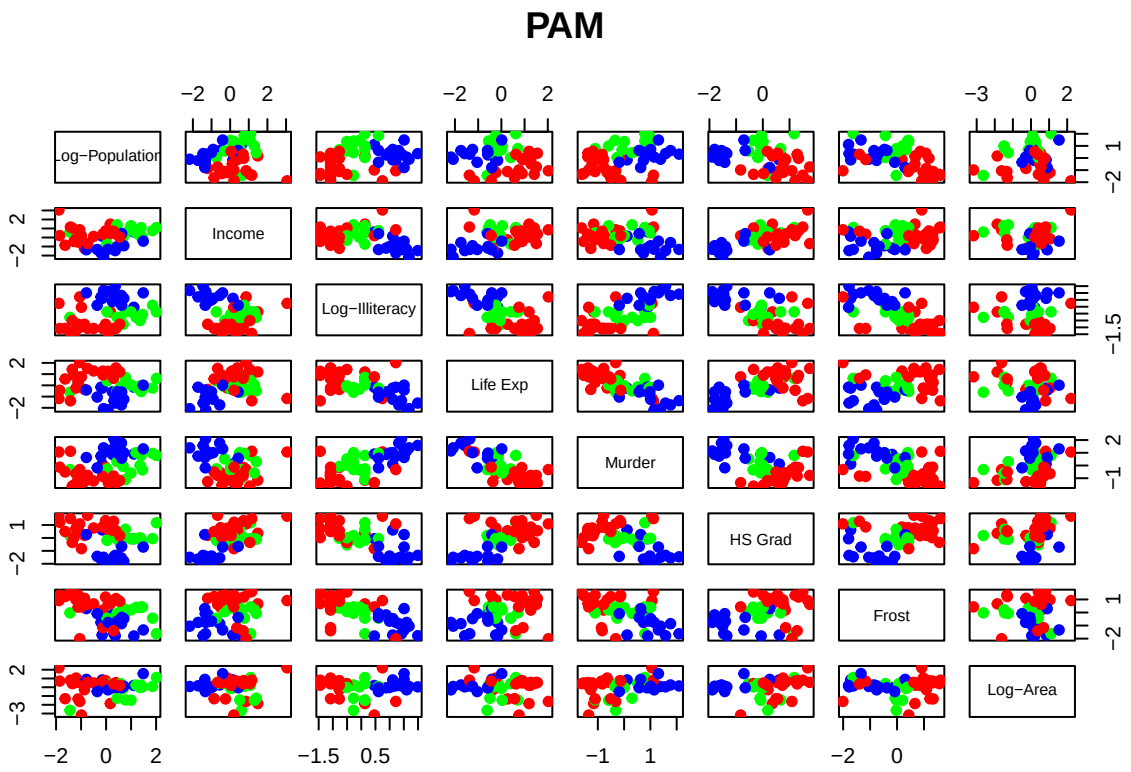
Extraemos los clusters que están dentro del algoritmo PAM y los visualizamos

```
cl.pam<-pam.3$clustering
cl.pam
```

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

#5.- Scatter plot de la matriz con los grupos

```
col.cluster<-c("blue","red","green")[cl.pam]
pairs(X.s, col=col.cluster, main="PAM", pch=19)
```



Este gráfico es muy parecido al de k-means pues están agrupadas las observaciones con una correlación muy parecida;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.

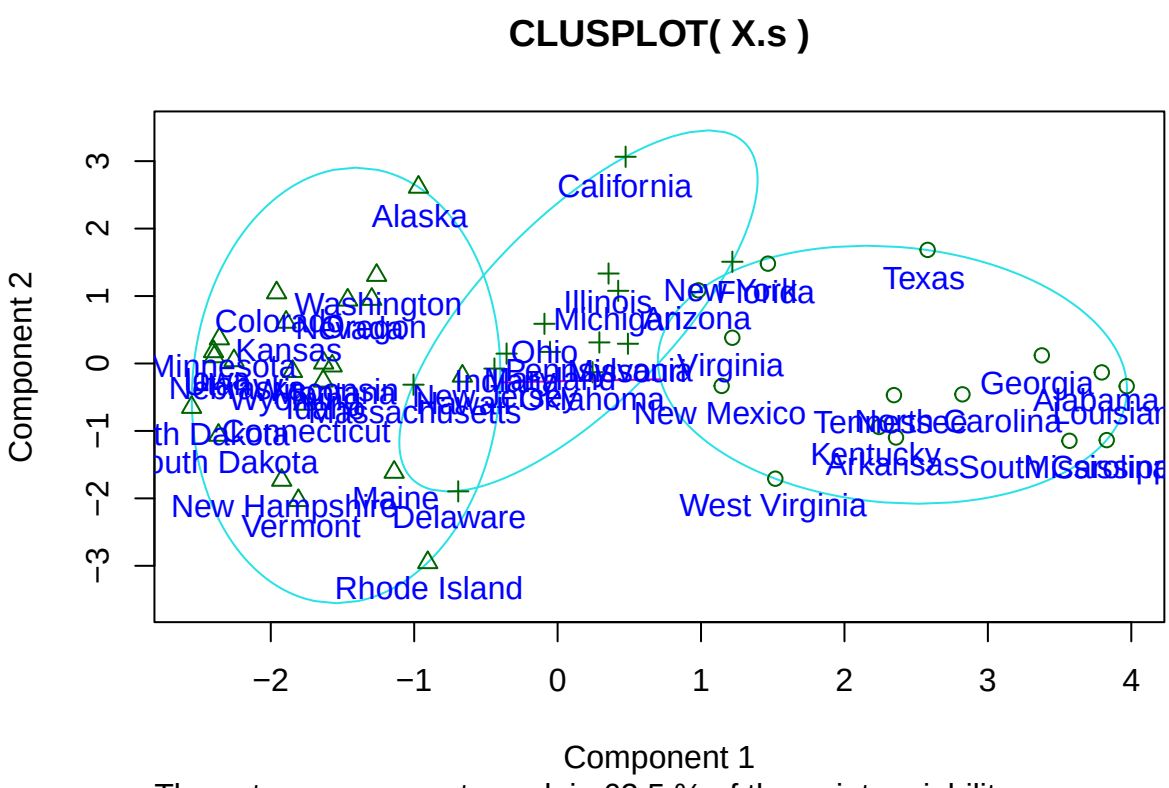
Visualizacion con las dos componentes principales

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

librería necesaria

```
library(cluster)
```

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



Podemos ver gráficamente la agrupación en clusters y saber cual clusters es cada uno con ayuda de la lista de agrupamiento y la figura de cada agrupación podemos decir el grupo al que pertenece, en el que esta Texas es el cluster 1, California cluster 3 y Alaska cluster 2 cambia a difencia del metodo o algoritmo K-means.

También nos indica que en estos dos componentes principales se explica el 62.5% de la variabilidad podemos decir que es buena y es practicamente como el de k-means .

Silhouette

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

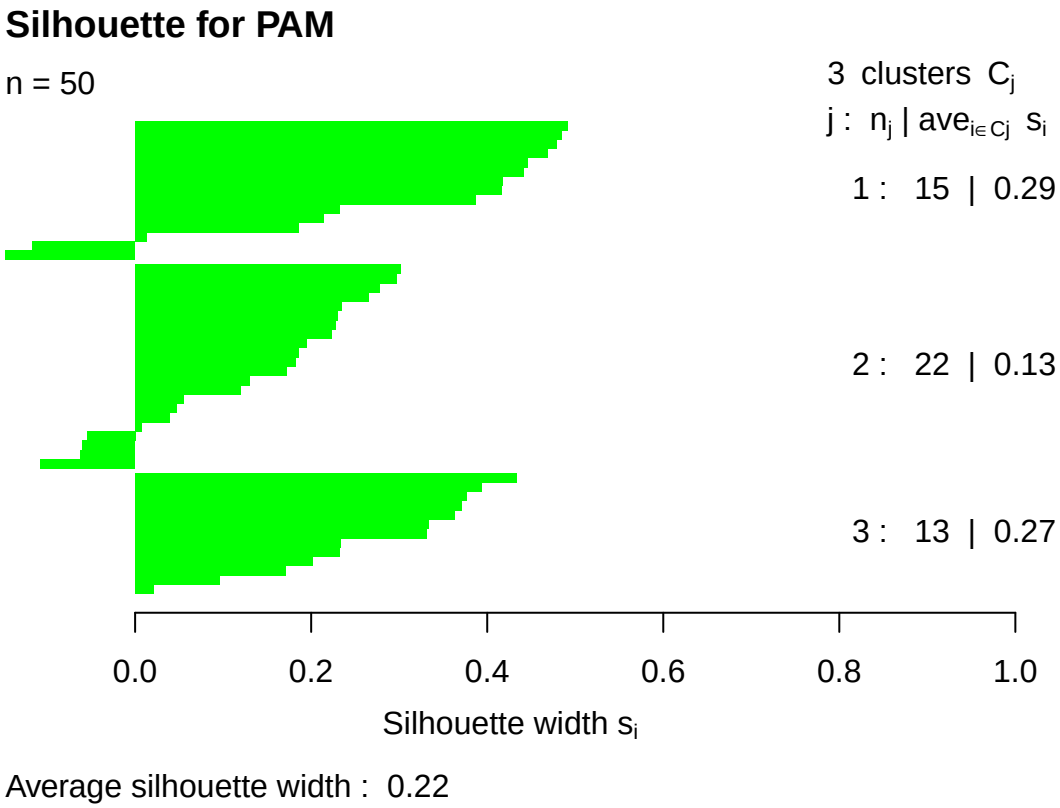
Generación de los cálculos

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.pam<-silhouette(cl.pam, dist.Euc)
```

Generación del gráfico

```
plot(Sil.pam, main="Silhouette for PAM",
     col="green")
```



Interpretación

Se puede ver que el ancho para cada silhouette por cluster, es medianamente bueno en el 1 y 3 pero en el uno hay observaciones negativas y en el 2 es más bajo a comparación de los otros dos además de presenta más, de una observación negativa, esto puede ser por que en el agrupamiento algunas observaciones se traslapan en los grupos, también que el algoritmo PAM es utilizado en presencia de datos atipicos pues es más robusto que k-means

Adicional el **average silhouette width**(ancho medio del silhouette)=0.22 no es muy alto debería ser mejor pero como es un algoritmo robusto baja la precisión; este cifra es más baja que en el silhouette de k-means.

Ejercicio

1. replicar el script pero con un numero de clusters diferentes a 3 y 1
2. incluir la interpretación del silhouette

Tratare de sintetizar sin explicar ,os pasos para que no sea redundante el documento.

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 datos

Transformación de las variables x1,x3 y x8 con la función 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 PAM

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)
```

3.- Aplicacion del algoritmo

Aquí aplicamos el algoritmo PAM con dos grupos como en el de K-means a la matriz de las variables escaladas.

```
Pam.3<-pam(x.s,2)
```

4.- Clusters

```
Cl.pam<-Pam.3$clustering
Cl.pam
```

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

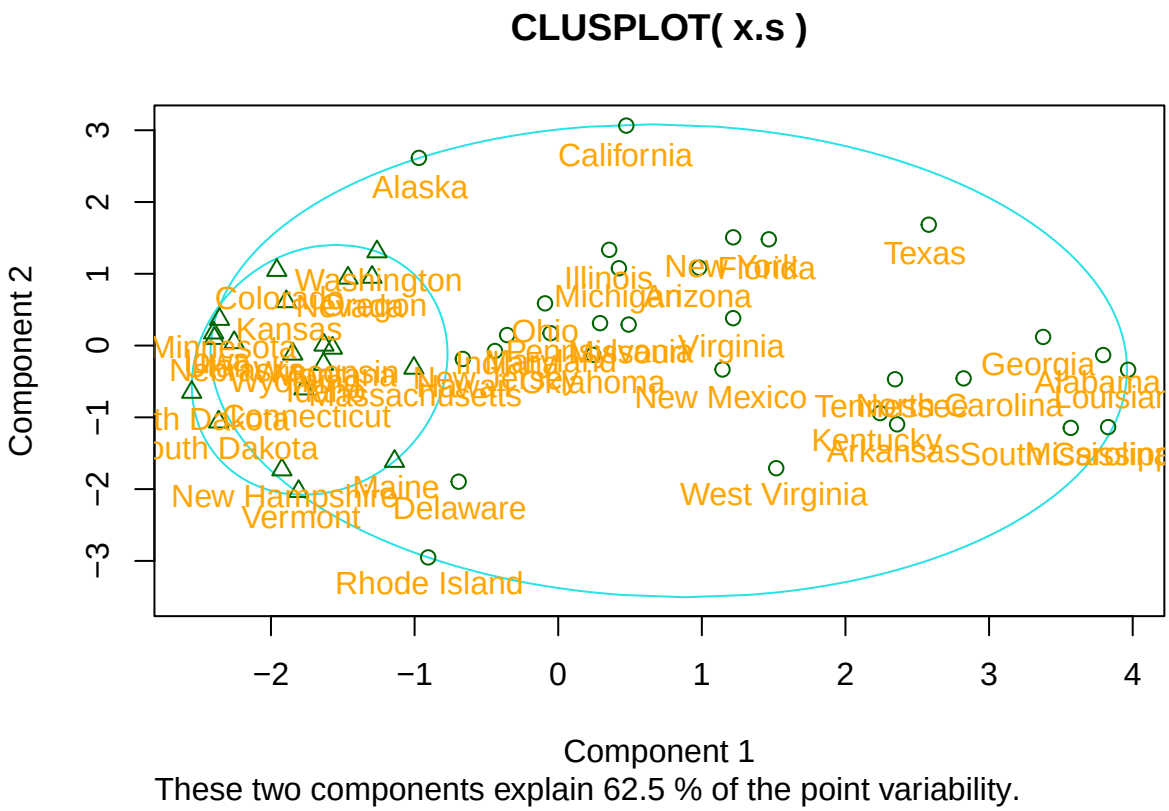
Visualizacion con las dos componentes principales

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

librería necesaria

```
library(cluster)
```

```
clusplot(x.s,Cl.pam)
text(princomp(x.s)$scores[,1:2],
     labels=rownames(x.s),pos=1, col="orange")
```



Podemos ver gráficamente la agrupación en clusters y saber cual clusters es cada uno con ayuda de la lista de agrupamiento y la figura de cada agrupación podemos decir el grupo al que pertenece, en el que esta California es el cluster 1 y Washington es el cluster 2 pero esta dentro del cluster 1 prácticamente.

También nos indica que en estos dos componentes principales se explica el 62.5% de la variabilidad podemos decir que es buena y es prácticamente como el de k-means e igual al del tres clusters.

Silhouette

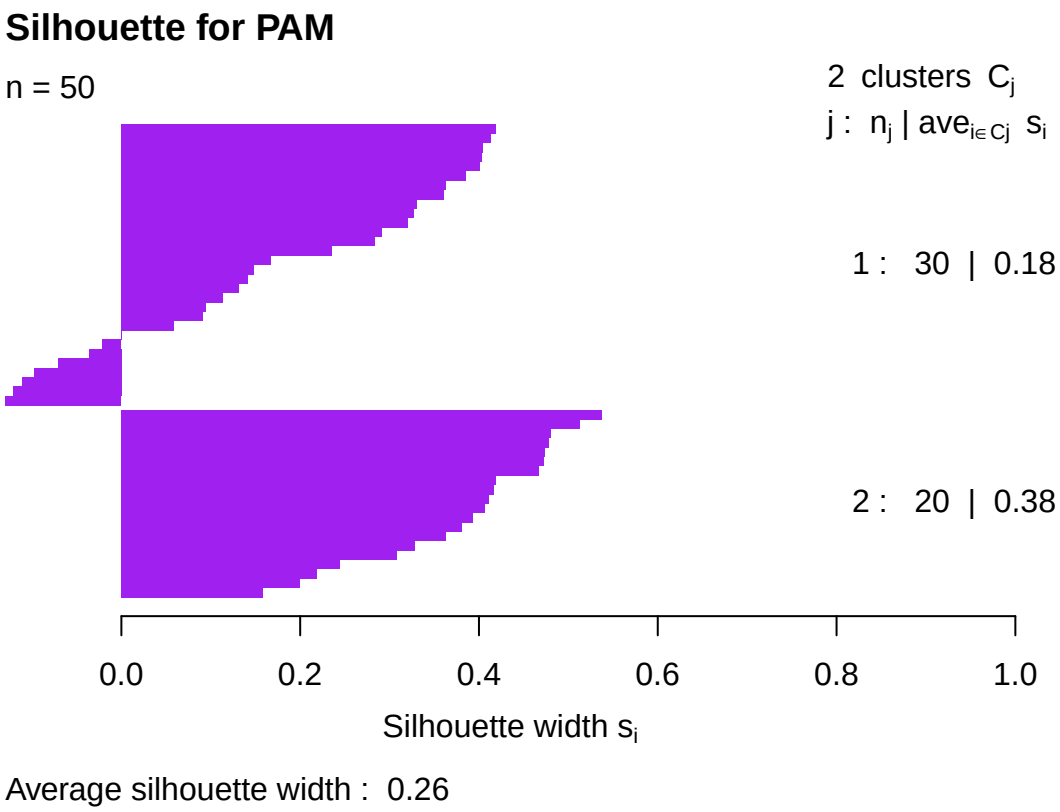
Representación gráfica de la eficacia de clasificación de una observación dentro de un grupo.

Generacion de los calculos

```
dist.Euc<-dist(x.s, method = "euclidean")
sil.pam<-silhouette(Cl.pam, dist.Euc)
```

Generacion del grafico

```
plot(sil.pam, main="Silhouette for PAM",
     col="purple")
```



Con dos clusters podemos tener un buen ancho de Silhouette para cada cluster. Aunque en el primer cluster hay observaciones negativas el averge del Silhouette general es mayor que si elijo 15 clusters cabe destacar que realice simulaciones con distintas cantidades de clusters pero no son incluidas por que seria mucho y sin sentido, aunque el averege general mejora en diferentes cantidades de clusters, el ancho de Silhouette para cada cluster puede ser un poco mejor en algunos, en otros es pésimo así que es mejor usar dos pues esta más balanceado.