k-means

Medel Colorado Yoselin Merari

2022-06-05

K-MEANS

Cargar la matriz de datos.

Se cnsieran las medianas y busca k objetos representativos

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

#X #————————————————- # Transformacion de datos #————————————————-

#1.- 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"
```

#———————————————— # Método k-means #————————————————

#1.- Separación de filas y columnas.

```
dim(X)
```

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

```
n<-dim(X)[1]
p<-dim(X[2])
```

2.- Estandarización univariante.

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

3.- Algoritmo k-medias (6 grupos)

nstar es cantidad de subconjuntos aleatorios que se escogen

para realizar los cálculos de algoritmo.

el 3 es el número de clouster o de agrpupaciones, en este caso se utilizan 3

```
Kmeans.6<-kmeans(X.s, 6, nstart=25)
```

centroides

```
Kmeans.6$centers
```

```
##   Log-Population     Income Log-Illiteracy   Life Exp     Murder     HS Grad
## 1     1.05203572  0.2689748      0.1658871 -0.1124169  0.4831422 -0.06765652
## 2    -0.02012796  0.2632441     -1.0527537  1.1656294 -0.9511840  0.92206977
## 3    -1.30355300 -0.2681986     -0.9775813  0.3548885 -0.9218376  0.46019574
## 4     0.12233125 -1.3014617      1.3019262 -1.1773136  1.0919809 -1.41578257
## 5    -0.15758822  0.9109826      0.2165582  0.5182427 -0.6480455  0.18472210
## 6    -1.65470747  2.1094604     -0.3490974 -1.2728011  1.0895183  1.58994719
##        Frost    Log-Area
## 1 -0.4380016  0.37632593
## 2  0.3010938  0.49075236
## 3  1.1526361  0.03872450
## 4 -0.7206500  0.07602772
## 5 -0.1187800 -1.92526117
## 6  1.2608490  1.51085951
```

cluster de pertenencia

```
Kmeans.6$cluster
```

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

```
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              3              4              1              2              3
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              1              2              4              2              3
```

4.- SCDG

#hasta aquí llego el minimo de scdg la idea es llegar a 0

```
SCDG<-sum(Kmeans.6$withinss)
SCDG
```

```
## [1] 121.0769
```

5.- Clusters

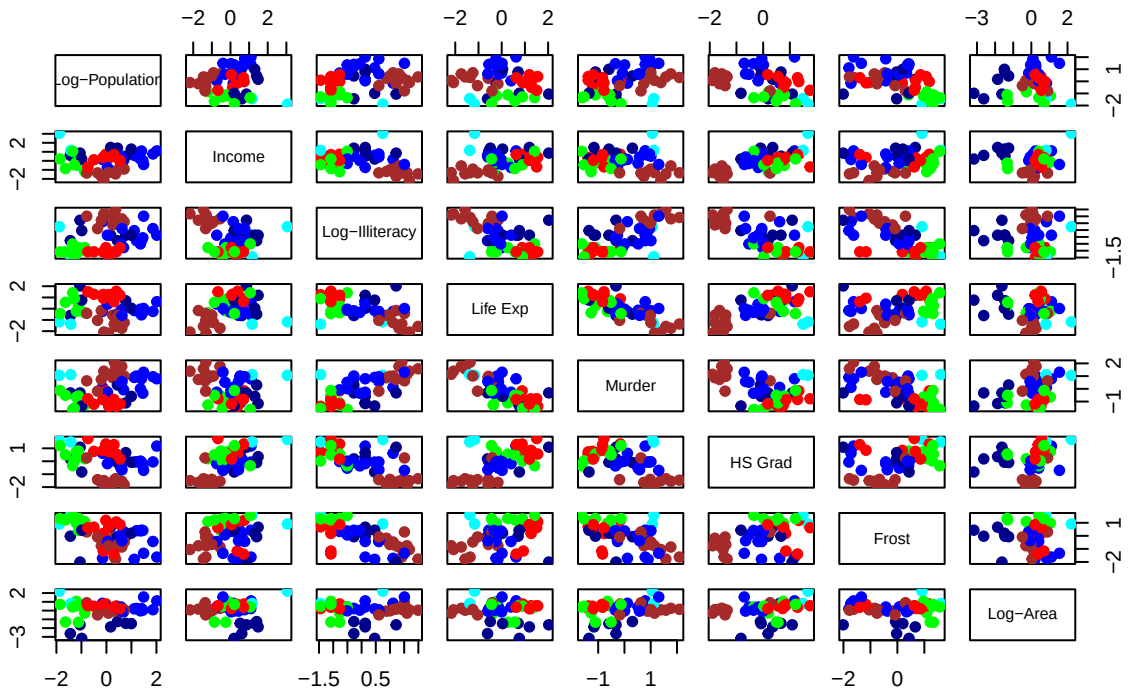
```
cl.kmeans<-Kmeans.6$cluster
cl.kmeans
```

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

6.- Scatter plot con la division de grupos

obtenidos (se utiliza la matriz de datos centrados)

```
col.cluster<-c("blue", "red", "green","brown","darkblue","cyan")[cl.kmeans]
pairs(X.s, col=col.cluster, main="k-means", pch=19)
```



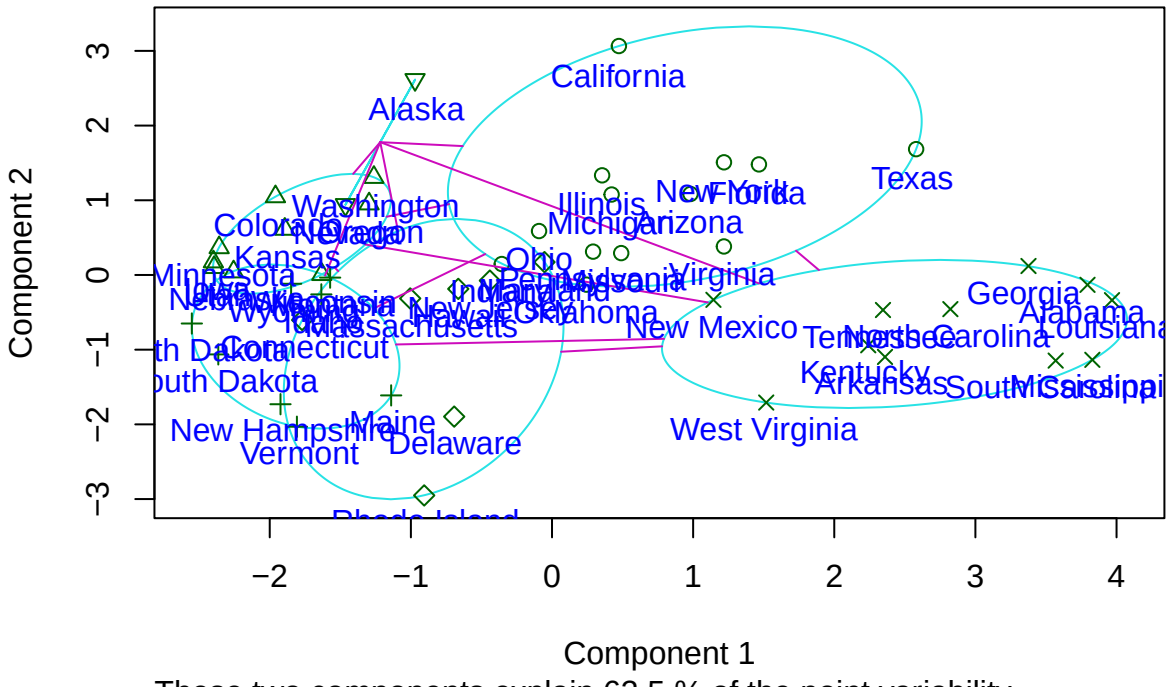
#———————————————— # Visualización con las dos componentes principales #————————————————

Instalar apaquete

```
install.packages("cluster")
library(cluster)
```

Gráfico

```
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")
```

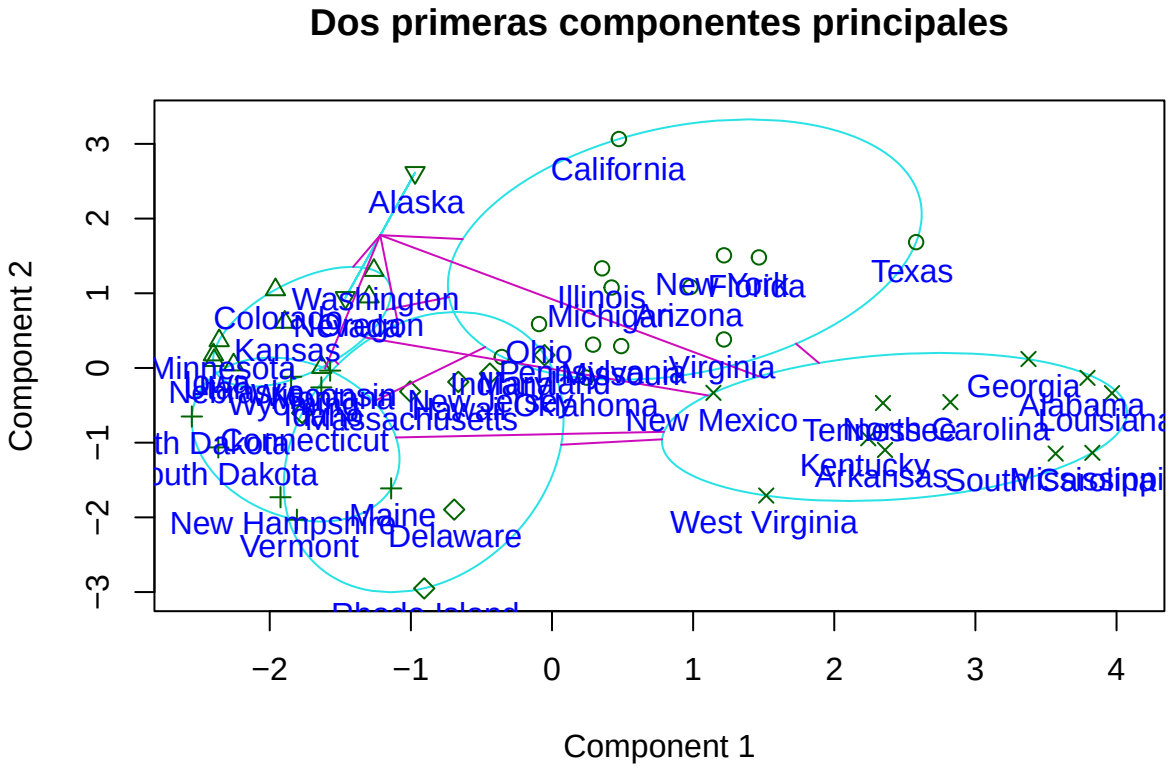


These two components explain 62.5 % of the point variability.

Visualización con las dos componentes principales

Gráfico

```
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.

De aqui se puede tomar la descicio para aumentar el numero de clousters

#———————————- # Silhouette #———————————— # Representacion gráfica de la eficacia de # clasificación de una observación dentro de un # grupo.

1.- Generación de los cálculos

```
dist.Euc<-dist(X.s, method = "euclidean")
```

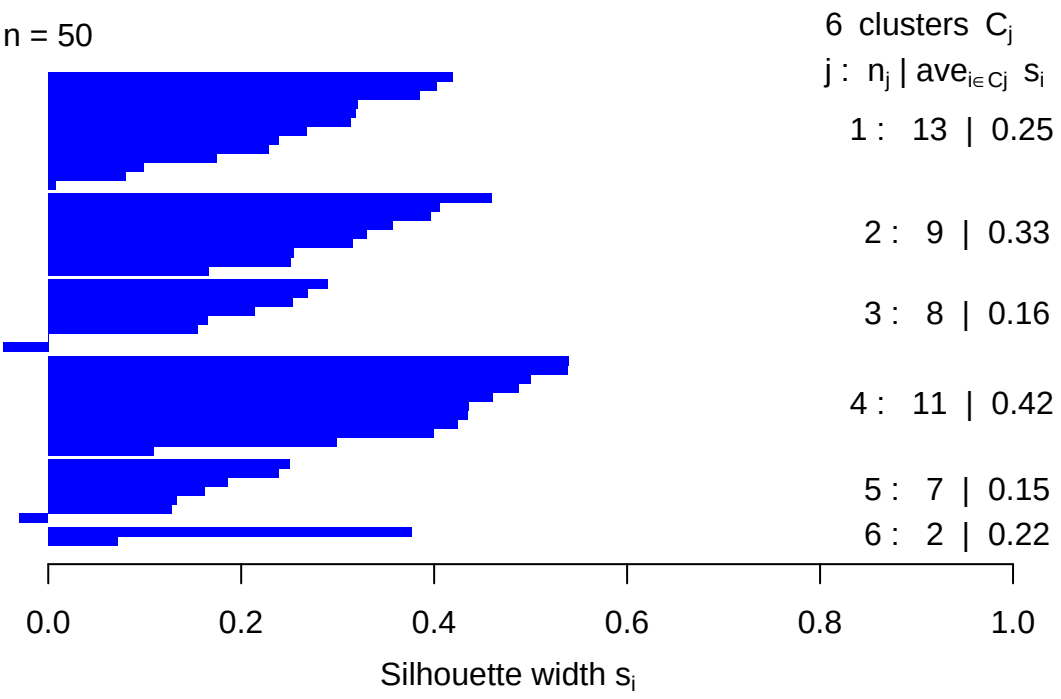
El cl.kmeans es dode se se encuentran los closters

```
Sil.kmeans<-silhouette(cl.kmeans, dist.Euc)
```

#2.- Generación del gráfico

Los ultimos números de la derecha son la probabilidad si es bajo es decir que la clasificacion es baja

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



Average silhouette width : 0.27