Análisis Factorial

Oscar Elí Bonilla Morales

2022-05-20

El análisis factorial es una técnica de reducción estadística el cual tiene como objetivo explicar las posibles correlaciones entre variables especificas, para esto es necesario tener en cuenta el efecto de otras variables, o los factores, que no son observables.

Para conseguir esto lo que se busca es reducir las dimensiones de nuestra matriz a un tamaño más manejable. Para conseguir esto, se utilizan una serie de combinaciones lineales de las observadas con otras que no son visibles.

Ejemplo

Para este ejemplo, utilizaremos una matriz precargada en la base de R llamada "state.x77"

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

```
##            Population Income Illiteracy Life Exp Murder HS Grad Frost   Area
## Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
## Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
## Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
## California      21198   5114        1.1    71.71   10.3    62.6    20 156361
## Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
```

Esta matriz cuenta con datos sociodemográficos referente a los distintos estados de estados unidos, algunas de las variables son el ingreso, el numero de poblacion, la expectativa de vida, asesinatos, entre otras.

Quitar los espacios de los nombres

Es necesario eliminar los espacios en los nombres de las variables, ya que al momento de realizar este tipo de análisis puede generar conflictos

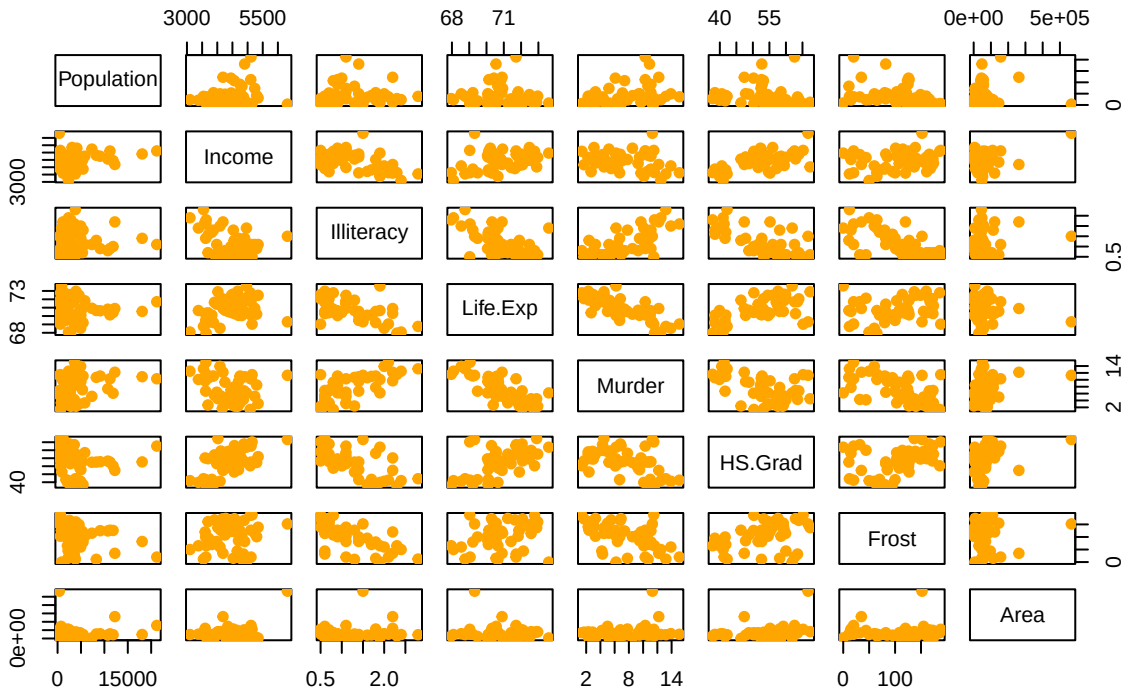
```
colnames(x)[4]="Life.Exp"
colnames(x)[6]= "HS.Grad"
```

Separa n (estados) y p (variables)

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

Generacion de un scater plot de las variables originales

```
pairs(x, col="orange", pch=19, main="matriz original")
```



Se realiza la visuaizacion de los datos originales para explorar y observar el comportamiento de algunas variables.

Transformación de alguna varibles

Aplicación de logaritmo para las columnas 1,3 y 8

Para este caso realizamos logaritmo ya que los datos que se manejaran son valores muuy grandes.

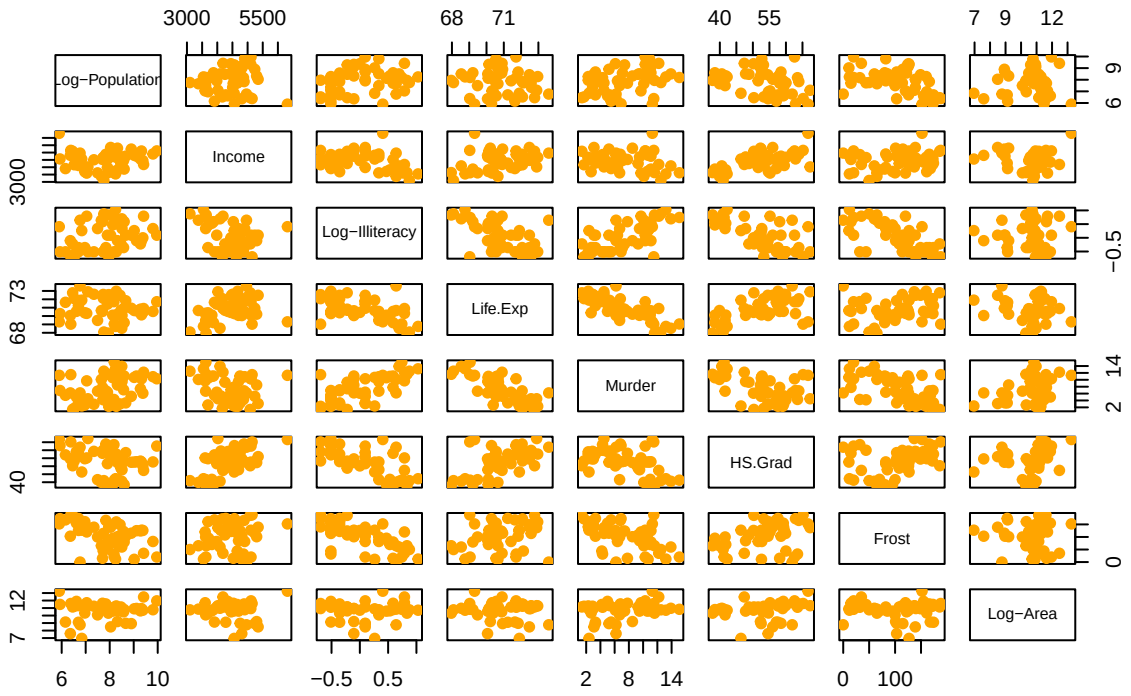
```
x[,1]<-log(x[,1])
colnames(x)[1]<-"Log-Population"

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

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

Grafico scaterplot de nuestra matriz original con 3 variables incluidas

```
pairs(x,col="orange", pch=19, main="Matriz original")
```



En este gráfico es posible observar un cambio significativo en la dispersión de algunas variables, por lo general, estas transformaciones se realizan si es que se desean obtener mejores resultados.

Nota:

Como las variables tiene diferentes unidades de medida, se va a implementar la matriz de correlaciones para estimar la matriz de carga.

Reduccion de la dimensionalidad

Análisis Factorial de componentes principales (PCFA)

Para poder realizar el (PCFA) de manera exitosa es necesario realizar una matriz de medias al igual que una de las correlaciones de nuestras variables.

Calcular la matriz de medias y de correlaciones

Matriz de medias

```
mu<-colMeans(x)
mu
```

```
## Log-Population         Income Log-Illiteracy       Life.Exp         Murder
##   7.863443e+00   4.435800e+03   3.128251e-02   7.087860e+01   7.378000e+00
##        HS.Grad          Frost       Log-Area
##   5.310800e+01   1.044600e+02   1.066237e+01
```

#Matriz de correlaciones

```
R<-cor(x)
R
```

```
##                Log-Population       Income Log-Illiteracy   Life.Exp     Murder
## Log-Population     1.00000000  0.034963788     0.28371749 -0.1092630  0.3596542
## Income             0.03496379  1.000000000    -0.35147773  0.3402553 -0.2300776
## Log-Illiteracy     0.28371749 -0.351477726     1.00000000 -0.5699943  0.6947320
## Life.Exp          -0.10926301  0.340255339    -0.56999432  1.0000000 -0.7808458
## Murder             0.35965424 -0.230077610     0.69473198 -0.7808458  1.0000000
## HS.Grad           -0.32211720  0.619932323    -0.66880911  0.5822162 -0.4879710
## Frost             -0.45809012  0.226282179    -0.67656232  0.2620680 -0.5388834
## Log-Area           0.08541473 -0.007462068    -0.05830524 -0.1086351  0.2963133
##                   HS.Grad       Frost     Log-Area
## Log-Population -0.3221172 -0.45809012  0.085414734
## Income          0.6199323  0.22628218 -0.007462068
## Log-Illiteracy -0.6688091 -0.67656232 -0.058305240
## Life.Exp        0.5822162  0.26206801 -0.108635052
## Murder         -0.4879710 -0.53888344  0.296313252
## HS.Grad         1.0000000  0.36677970  0.196743429
## Frost           0.3667797  1.00000000 -0.021211992
## Log-Area        0.1967434 -0.02121199  1.000000000
```

2.- Reducción de la dimensionalidad mediante Análisis factorial de componentes principales (PCFA).

Calcular los valores y vectores propios.

Utilizando nuestros *Valores orignales* calcularemos los valores propios de nuestra matriz de correlaciones.

```
eR<-eigen(R)
```

Valores propios

```
eR$values
```

```
## [1] 3.6796976 1.3201021 1.1357357 0.7517550 0.6168266 0.2578511 0.1366186
## [8] 0.1014132
```

Vectores propios

```
eR$vectors
```

```
##             [,1]        [,2]        [,3]       [,4]        [,5]       [,6]
## [1,] -0.23393451 -0.41410075  0.50100922  0.2983839  0.58048485  0.0969034
## [2,]  0.27298977 -0.47608715  0.24689968 -0.6449631  0.09036625 -0.3002708
## [3,] -0.45555443  0.04116196  0.12258370 -0.1824471 -0.32684654 -0.6084112
## [4,]  0.39805075 -0.04655529  0.38842376  0.4191134 -0.26287696 -0.3565095
## [5,] -0.44229774 -0.27640285 -0.21639177 -0.2610739  0.02383706  0.1803894
## [6,]  0.41916283 -0.36311753 -0.06807465 -0.1363534 -0.34015424  0.3960855
## [7,]  0.36358674  0.21893783 -0.37542494 -0.1299519  0.59896253 -0.3507630
## [8,] -0.03545293 -0.58464797 -0.57421867  0.4270918 -0.06252285 -0.3012063
##             [,7]        [,8]
## [1,] -0.1777562 -0.23622413
```

```
## [2,]  0.3285840  0.12483849
## [3,] -0.3268997 -0.39825363
## [4,] -0.3013983  0.47519991
## [5,] -0.4562245  0.60970476
## [6,] -0.4808140 -0.40675672
## [7,] -0.4202943 -0.06001175
## [8,]  0.2162424 -0.05831177
```

Al realizar esta funcion es posible observar que obtenemos la matriz de autovalores al igual que la de autovectores, por lo que será necesario separarlas.

Valores propios

```
eigen.val<-eR$values
eigen.val
```

```
## [1] 3.6796976 1.3201021 1.1357357 0.7517550 0.6168266 0.2578511 0.1366186
## [8] 0.1014132
```

Vectores propios

```
eigen.vec<-eR$vectors
eigen.vec
```

```
##             [,1]        [,2]        [,3]       [,4]        [,5]       [,6]
## [1,] -0.23393451 -0.41410075  0.50100922  0.2983839  0.58048485  0.0969034
## [2,]  0.27298977 -0.47608715  0.24689968 -0.6449631  0.09036625 -0.3002708
## [3,] -0.45555443  0.04116196  0.12258370 -0.1824471 -0.32684654 -0.6084112
## [4,]  0.39805075 -0.04655529  0.38842376  0.4191134 -0.26287696 -0.3565095
## [5,] -0.44229774 -0.27640285 -0.21639177 -0.2610739  0.02383706  0.1803894
## [6,]  0.41916283 -0.36311753 -0.06807465 -0.1363534 -0.34015424  0.3960855
## [7,]  0.36358674  0.21893783 -0.37542494 -0.1299519  0.59896253 -0.3507630
## [8,] -0.03545293 -0.58464797 -0.57421867  0.4270918 -0.06252285 -0.3012063
##            [,7]        [,8]
## [1,] -0.1777562 -0.23622413
## [2,]  0.3285840  0.12483849
## [3,] -0.3268997 -0.39825363
## [4,] -0.3013983  0.47519991
## [5,] -0.4562245  0.60970476
## [6,] -0.4808140 -0.40675672
## [7,] -0.4202943 -0.06001175
## [8,]  0.2162424 -0.05831177
```

Calcular la proporcion de variabilidad

se obtiene diviendo los valores propios entre la suma de los valores propios

```
prop.var<-eigen.val/sum(eigen.val)
prop.var
```

```
## [1] 0.45996220 0.16501277 0.14196697 0.09396938 0.07710332 0.03223139 0.01707733
## [8] 0.01267665
```

Calcular la proporcion de variabilidad acumulada

```
prop.var.acum<-cumsum(eigen.val)/sum(eigen.val)
prop.var.acum
```

```
## [1] 0.4599622 0.6249750 0.7669419 0.8609113 0.9380146 0.9702460 0.9873233
## [8] 1.0000000
```

En este caso, escogeremos los 3 primeros factores, ya que son los que mejor representan nuestra variabilidad acumulada

0.4478068 0.6345629 0.7696234

Estimacion de la matriz de carga

Nota:

se estima la matriz de carga usando los autovalores y autovectores, despues, se aplica la rotación varimax.

Primera estimación de Lamda mayuscula

Se calcula multiplicando la matriz de los3 primeros autovectores por la matriz diagonal formada por la raiz cuadrada de los primeros 3 autovalores.

```
L.est.1<-eigen.vec[,1:3] %*% diag(sqrt(eigen.val[1:3]))
L.est.1
```

```
##             [,1]        [,2]        [,3]
## [1,] -0.44874575 -0.47578394  0.53393005
## [2,]  0.52366367 -0.54700365  0.26312322
## [3,] -0.87386900  0.04729332  0.13063856
## [4,]  0.76356236 -0.05349003  0.41394671
## [5,] -0.84843932 -0.31757498 -0.23061066
## [6,]  0.80406070 -0.41720642 -0.07254777
## [7,]  0.69745163  0.25155014 -0.40009375
## [8,] -0.06800771 -0.67173536 -0.61195003
```

Rotación varimax

```
L.est.1.var<-varimax(L.est.1)
L.est.1.var
```

```
## $loadings
##
## Loadings:
##      [,1]   [,2]   [,3]
## [1,]                 0.840
## [2,]  0.785 -0.106  0.121
## [3,] -0.665         0.583
## [4,]  0.763  0.384 -0.168
## [5,] -0.573 -0.528  0.517
## [6,]  0.825 -0.202 -0.323
## [7,]  0.281        -0.794
## [8,]        -0.906
##
```

```
##                  [,1]  [,2]  [,3]
## SS loadings    2.744 1.300 2.091
## Proportion Var 0.343 0.163 0.261
## Cumulative Var 0.343 0.506 0.767
##
## $rotmat
##            [,1]      [,2]       [,3]
## [1,]  0.7824398 0.1724744 -0.5983649
## [2,] -0.5274231 0.6944049 -0.4895169
## [3,]  0.3310784 0.6986089  0.6342970
```

En el apartado de "loadings" es posible apreciar que la variable 1 tiene mayor carga en el factor 3, la variable 2 carga mas en el factor 1, la variable 3 carga negativo en el factor 1, en el factor 2 no carga y la mejor carga se presenta en el factor 3, asi es posible ver las mejores cargas entre variables y factores.

En el apartado de "ss loadings" se puede apreciiar como es que la proporciónn de la variabilidad para el factor 1 es de 0.33% para el 2 es de 0.16% y para el factor 3 es de 0.26% por lo que podemos considerar al factor 1 y al 3 ya que son los cuales nos otorgan mayor variabilidad.

Estimación de la matriz de los errores

Esta matriz tambien es conocida como la matriz de perturbaciones.

```
Psi.est.1<-diag(diag(R-as.matrix(L.est.1.var$loadings)%*% t(as.matrix(L.est.1.var$loadings))))
Psi.est.1
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]     [,6]      [,7]
## [1,] 0.2871756 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
## [2,] 0.0000000 0.3573295 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
## [3,] 0.0000000 0.0000000 0.2170499 0.0000000 0.0000000 0.000000 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 0.2427595 0.0000000 0.000000 0.0000000
## [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.1261156 0.000000 0.0000000
## [6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.174162 0.0000000
## [7,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.2902087
## [8,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
##           [,8]
## [1,] 0.0000000
## [2,] 0.0000000
## [3,] 0.0000000
## [4,] 0.0000000
## [5,] 0.0000000
## [6,] 0.0000000
## [7,] 0.0000000
## [8,] 0.1696637
```

Se utiliza el método Análisis de factor principal (PFA) para estimación de autovalores y autovectores

```
RP<-R-Psi.est.1
RP
```

```
##                Log-Population       Income Log-Illiteracy   Life.Exp     Murder
## Log-Population     0.71282441  0.034963788     0.28371749 -0.1092630  0.3596542
## Income             0.03496379  0.642670461    -0.35147773  0.3402553 -0.2300776
## Log-Illiteracy     0.28371749 -0.351477726     0.78295012 -0.5699943  0.6947320
```

```
## Life.Exp          -0.10926301  0.340255339    -0.56999432  0.7572405 -0.7808458
## Murder             0.35965424 -0.230077610     0.69473198 -0.7808458  0.8738844
## HS.Grad           -0.32211720  0.619932323    -0.66880911  0.5822162 -0.4879710
## Frost             -0.45809012  0.226282179    -0.67656232  0.2620680 -0.5388834
## Log-Area           0.08541473 -0.007462068    -0.05830524 -0.1086351  0.2963133
##                   HS.Grad       Frost     Log-Area
## Log-Population -0.3221172 -0.45809012  0.085414734
## Income          0.6199323  0.22628218 -0.007462068
## Log-Illiteracy -0.6688091 -0.67656232 -0.058305240
## Life.Exp        0.5822162  0.26206801 -0.108635052
## Murder         -0.4879710 -0.53888344  0.296313252
## HS.Grad         0.8258380  0.36677970  0.196743429
## Frost           0.3667797  0.70979126 -0.021211992
## Log-Area        0.1967434 -0.02121199  0.830336270
```

Calculo de la matriz de autovalores y autovectores

```
eRP<-eigen(RP)
```

Autovalores

```
eigen.val.RP<-eRP$values
eigen.val.RP
```

```
## [1]  3.46137648  1.10522195  0.88152416  0.48705680  0.35360597  0.02813553
## [7] -0.06758176 -0.11380367
```

Autovectores

```
eigen.vec.RP<-eRP$vectors
eigen.val.RP
```

```
## [1]  3.46137648  1.10522195  0.88152416  0.48705680  0.35360597  0.02813553
## [7] -0.06758176 -0.11380367
```

Proporcion de variabilidad

```
prop.var.RP<-eigen.val.RP/ sum(eigen.val.RP)
prop.var.RP
```

```
## [1]  0.564152306  0.180134556  0.143675179  0.079382934  0.057632455
## [6]  0.004585668 -0.011014811 -0.018548286
```

Proporcion de variabilidad acumulada

```
prop.var.RP.acum<-cumsum(eigen.val.RP)/ sum(eigen.val.RP)
prop.var.RP.acum
```

```
## [1] 0.5641523 0.7442869 0.8879620 0.9673450 1.0249774 1.0295631 1.0185483
## [8] 1.0000000
```

En este caso para la variabilidad acumulada escogeremos los 3 primeros factores *0.5641523*, *0.7442869*, *0.8879620*, ya que, aunque el 0.88 es un poco alto aun no es lo suficientemente alto como para descartarlo

De igual manera, pueden ser considerados solo 2 factores, todo depende de la manera en que esten distribuidos los datos. Para este caso escogeremos los 3 ya antes mencionados.

Estimación de la matriz de cargas con rotación varimax

```
L.est.2<-eigen.vec.RP[,1:3] %*% diag(sqrt(eigen.val.RP[1:3]))
L.est.2
```

```
##             [,1]        [,2]        [,3]
## [1,] -0.42621819 -0.27609775  0.56228420
## [2,]  0.48528446 -0.36092954  0.32467098
## [3,] -0.84791581  0.08163995  0.10816670
## [4,]  0.73812189  0.02688907  0.36866093
## [5,] -0.84699944 -0.34227865 -0.12211117
## [6,]  0.78817342 -0.40399024  0.04935203
## [7,]  0.66112453  0.12457105 -0.40191996
## [8,] -0.06868291 -0.77165602 -0.36531090
```

Rotacion varimax

```
L.est.2.var<-varimax(L.est.2)
L.est.2.var
```

```
## $loadings
##
## Loadings:
##      [,1]   [,2]   [,3]
## [1,]                0.756
## [2,]  0.677
## [3,] -0.651         0.560
## [4,]  0.748  0.303 -0.173
## [5,] -0.596 -0.477  0.517
## [6,]  0.803 -0.215 -0.309
## [7,]  0.285        -0.730
## [8,]        -0.854
##
##                  [,1]  [,2]  [,3]
## SS loadings    2.530 1.104 1.814
## Proportion Var 0.316 0.138 0.227
## Cumulative Var 0.316 0.454 0.681
##
## $rotmat
##            [,1]      [,2]       [,3]
## [1,]  0.7912457 0.1433911 -0.5944487
## [2,] -0.3865428 0.8705470 -0.3045202
## [3,]  0.4738301 0.4707302  0.7442434
```

Para esta matriz los valores otorgados en “Loadings” obtenemos que para la variable 1 no carga en absoluto en los factores 1 o 2 pero si en el 3, para la variable 2 solo carga en el primer factor, para la variable 3 el factor 1 carga de manera negativa, no carga en el factor 2 solo en el factor 3.

Prosiguiendo con el apartado “SS loadings” obtenemos que la proporción de la variabilidad es de 31% para el factor 1, de 13% para el factor 2 y 22% para el factor 3, por lo que podemos decir que los factores 1 y 3 son los que mejor representan la variabilidad.

Estimación de la matriz de covarianzas de los errores.

```
Psi.est.2<-diag(diag(R-as.matrix(L.est.2.var$loadings)%*% t(as.matrix(L.est.2.var$loadings))))
Psi.est.2
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,] 0.4259446 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.5288176 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000 0.2626737 0.0000000 0.0000000 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 0.3185422 0.0000000 0.0000000 0.0000000
## [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.1505261 0.0000000 0.0000000
## [6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.2131389 0.0000000
## [7,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.3858568
## [8,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##           [,8]
## [1,] 0.0000000
## [2,] 0.0000000
## [3,] 0.0000000
## [4,] 0.0000000
## [5,] 0.0000000
## [6,] 0.0000000
## [7,] 0.0000000
## [8,] 0.2663776
```

Obtencion de los scores de ambos métodos

PCFA

```
FS.est.1<-scale(x)%*% as.matrix(L.est.1.var$loadings)
head(FS.est.1)
```

```
##                  [,1]       [,2]      [,3]
## Alabama    -5.8407236 -1.3993672  4.000811
## Alaska      2.1244381 -3.6163397 -1.343594
## Arizona    -0.7724546 -1.1030150  1.786418
## Arkansas   -4.2696155 -0.1287634  1.868021
## California  1.5784398 -1.6386263  3.095976
## Colorado    3.3561948 -0.5747410 -1.995552
```

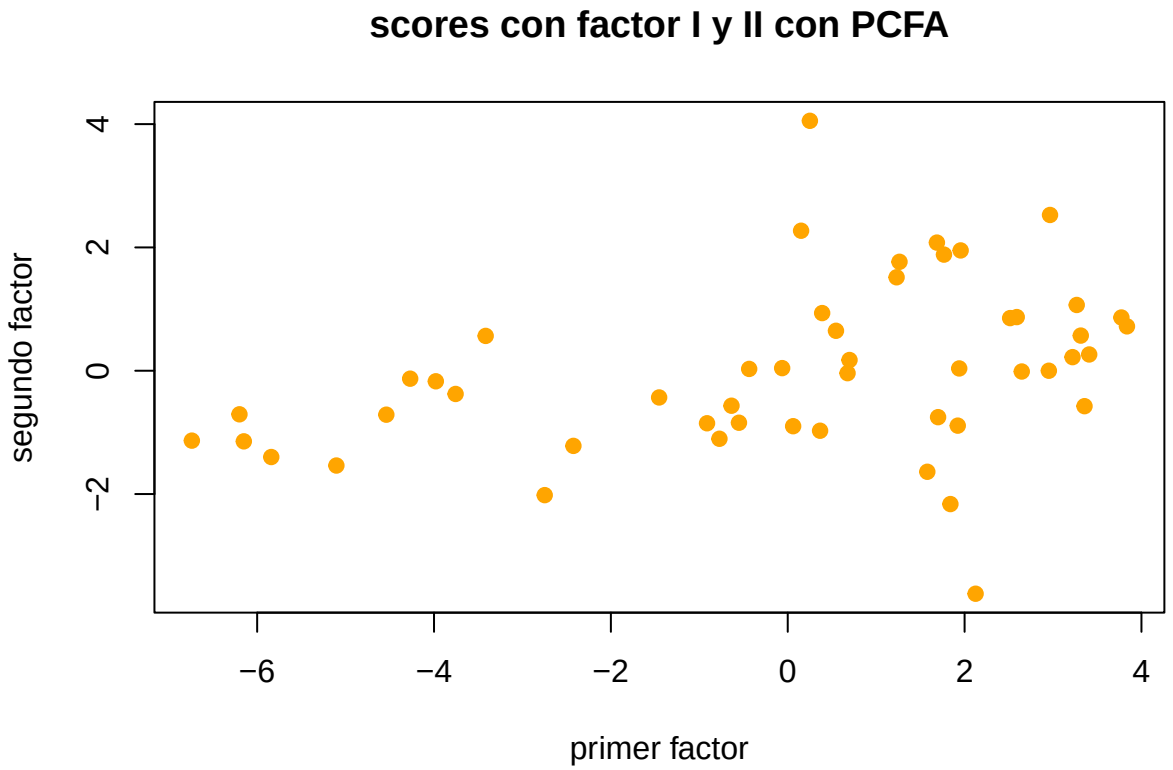
PFA

```
FS.est.2<-scale(x)%*% as.matrix (L.est.2.var$loadings)
head(FS.est.2)
```

```
##                  [,1]        [,2]      [,3]
## Alabama    -5.6976609 -1.13300587  3.903091
## Alaska      1.7792150 -3.31004955 -1.242553
## Arizona    -0.8094864 -1.00742357  1.683369
## Arkansas   -4.0445116 -0.03634031  1.889961
## California  1.2890077 -1.58952866  2.793822
## Colorado    3.2125676 -0.64509252 -1.910345
```

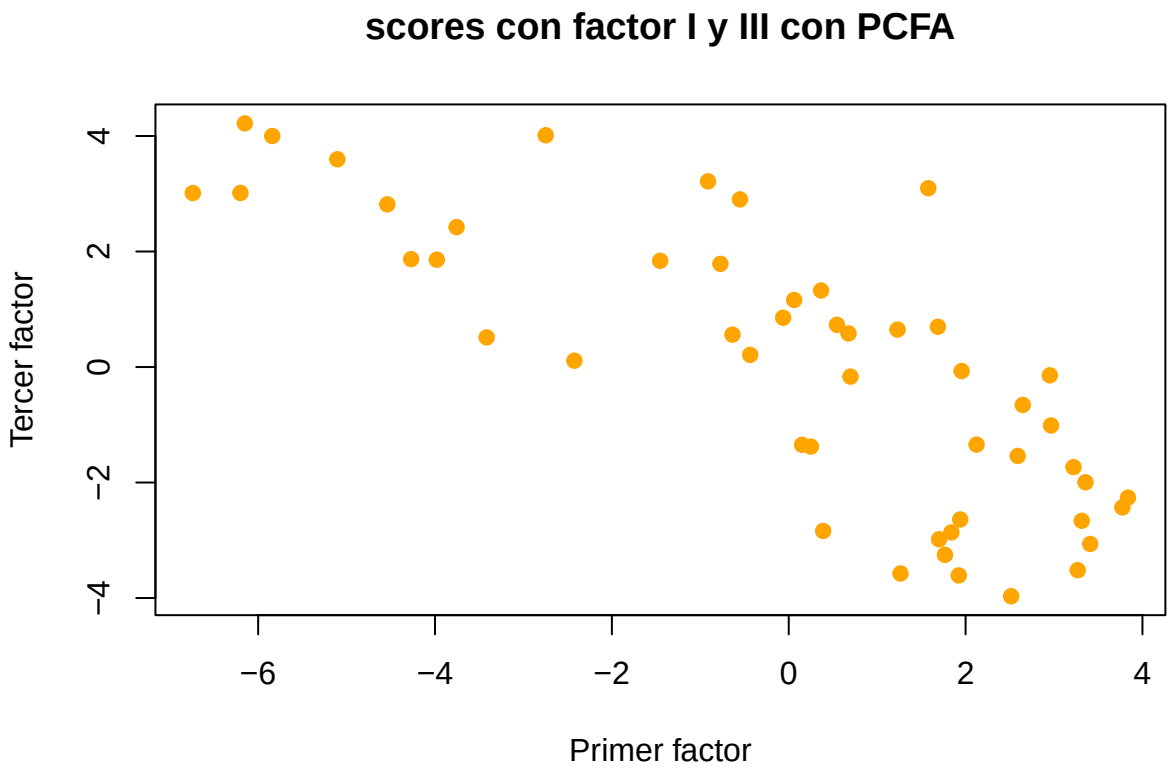
Factor I y II

```
pl1<-plot(FS.est.1[,1], FS.est.1[,2], xlab="primer factor",
     ylab="segundo factor", main="scores con factor I y II con PCFA",
     pch=19, col="orange")
```



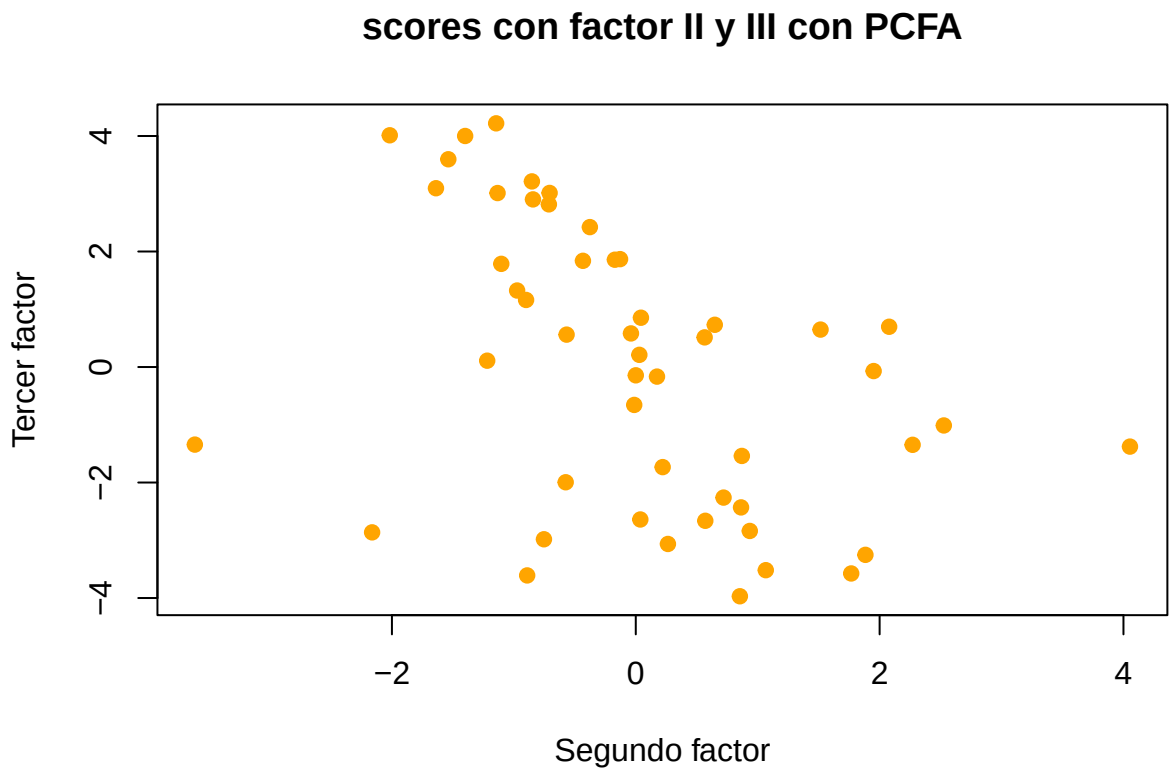
Factor I y III

```
pl2<-plot(FS.est.1[,1], FS.est.1[,3], xlab="Primer factor",
     ylab="Tercer factor", main="scores con factor I y III con PCFA",
     pch=19, col="orange")
```



Factor II y III

```
pl3<-plot(FS.est.1[,2], FS.est.1[,3], xlab="Segundo factor",
     ylab="Tercer factor", main="scores con factor II y III con PCFA",
     pch=19, col="orange")
```

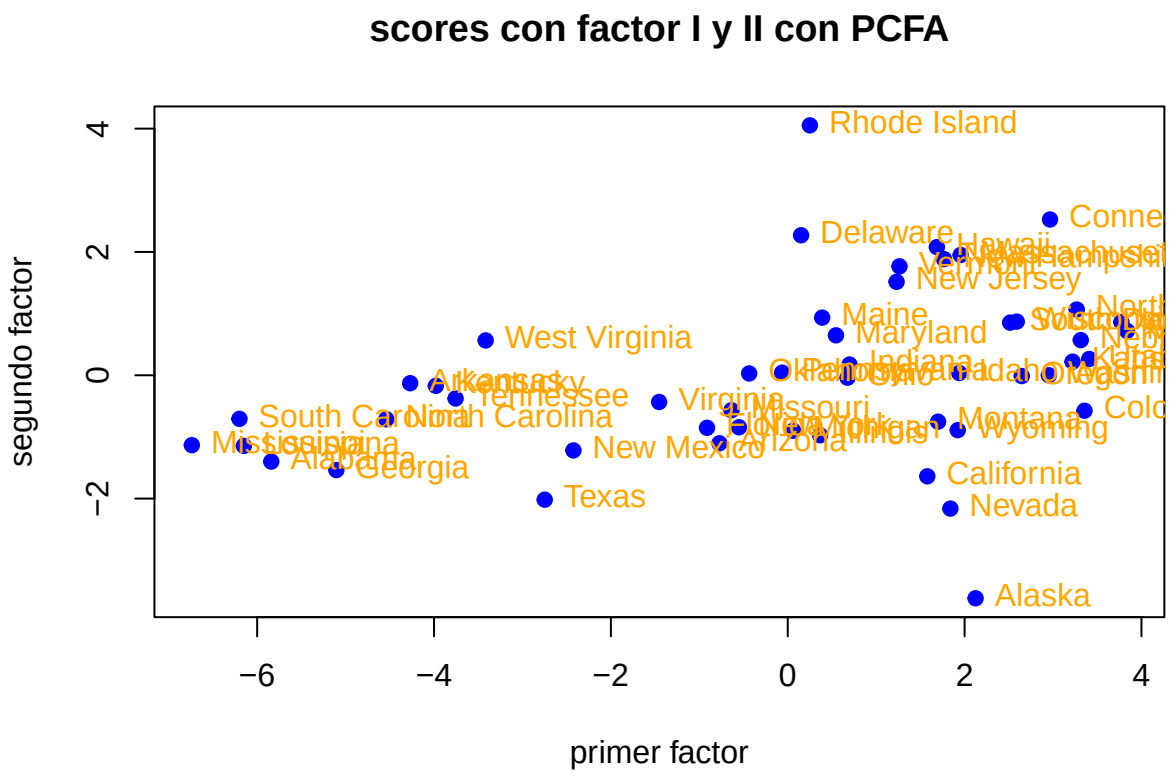


Gráficos con nombres de nuestras variables

```
par(mfrow=c(2,1))
```

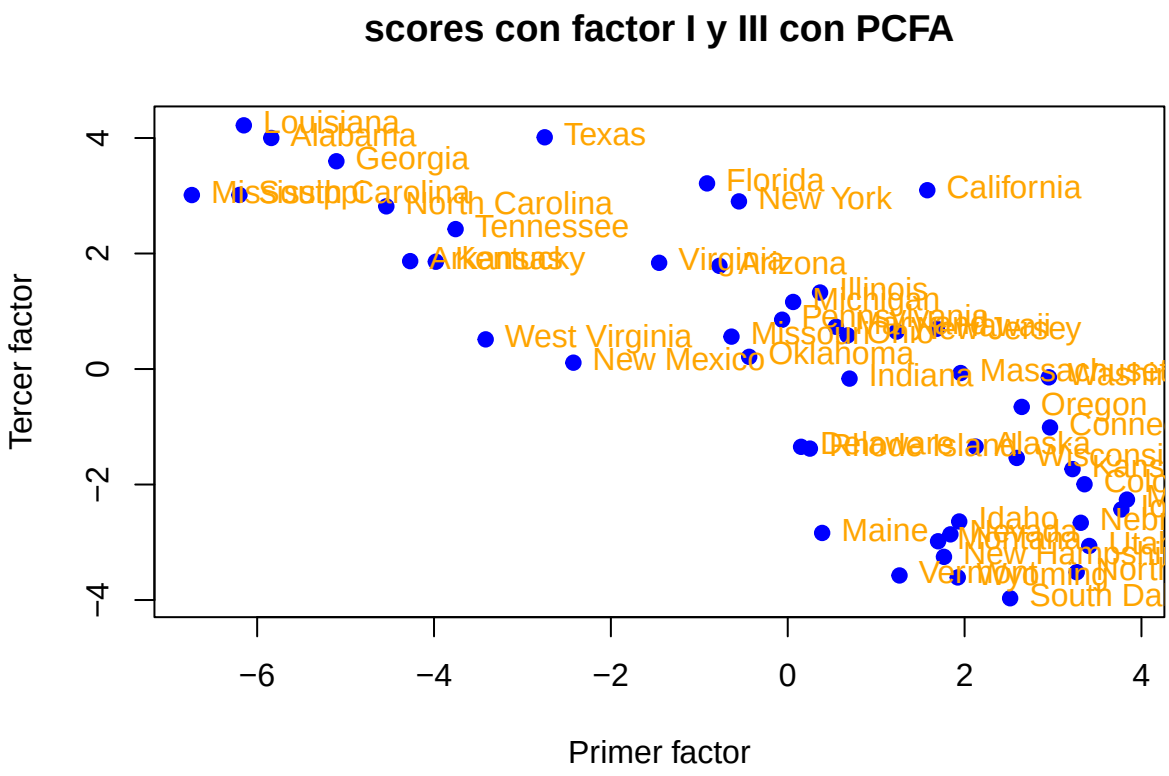
Factor I y II

```
pl1<-plot(FS.est.1[,1], FS.est.1[,2], xlab="primer factor",
          ylab="segundo factor", main="scores con factor I y II con PCFA",
          pch=19, col="blue")
text(FS.est.1[,1], FS.est.1[,2], labels = rownames(x), pos=4, col="orange")
```



Factor I y III

```
pl2<-plot(FS.est.1[,1], FS.est.1[,3], xlab="Primer factor",
          ylab="Tercer factor", main="scores con factor I y III con PCFA",
          pch=19, col="blue")
text(FS.est.1[,1], FS.est.1[,3], labels = rownames(x), pos=4, col="orange")
```


scores con factor I y III con PCFA
Tercer factor
Primer factor
Louisiana
Alabama
Georgia
Texas
Florida
New York
California
North Carolina
Tennessee
Virginia
Arizona
Illinois
West Virginia
New Mexico
Oklahoma
Indiana
Oregon
Alaska
Maine
Idaho
South Da
−6
−4
−2
0
2
4