Análisis factorial-bfi

Edgar Ortiz Mota

2022-05-14

Descarga de paquetes y librerias

```
install.packages("psych")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(psych)
install.packages("polycor")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(polycor)
```

```
##
## Attaching package: 'polycor'
```

```
## The following object is masked from 'package:psych':
##
##     polyserial
```

1.- Lectura de la matriz de datos

```
x1<-as.data.frame(bfi)
```

se seleccionaron las primeras 8 columnas

```
x= x1[1:200,1:8]
```

2.se eliminaron los NA de la matriz

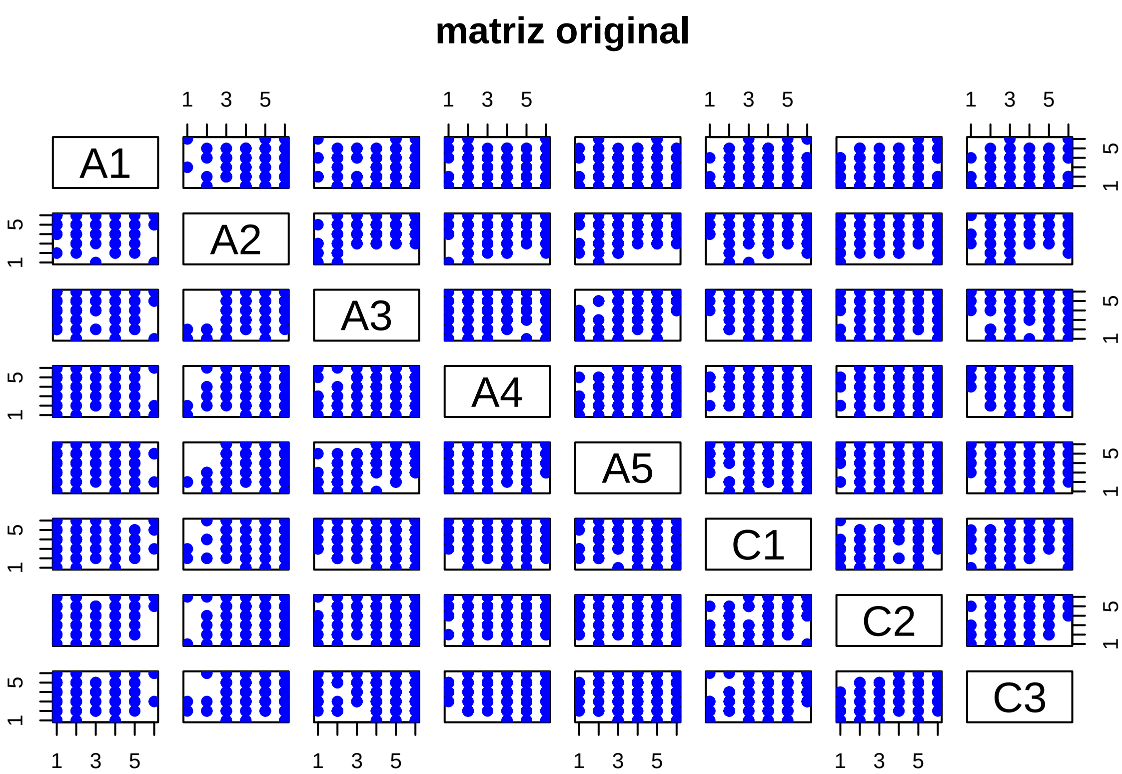
```
x <- na.omit(x)
```

3.- Separa n (estados) y p (variables)

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

4.- Generacion de un scater plot para la visualización de variables originales.

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



Transformación de alguna varibles

1.- Aplicamos logaritmo para las columnas 1,4 y 8

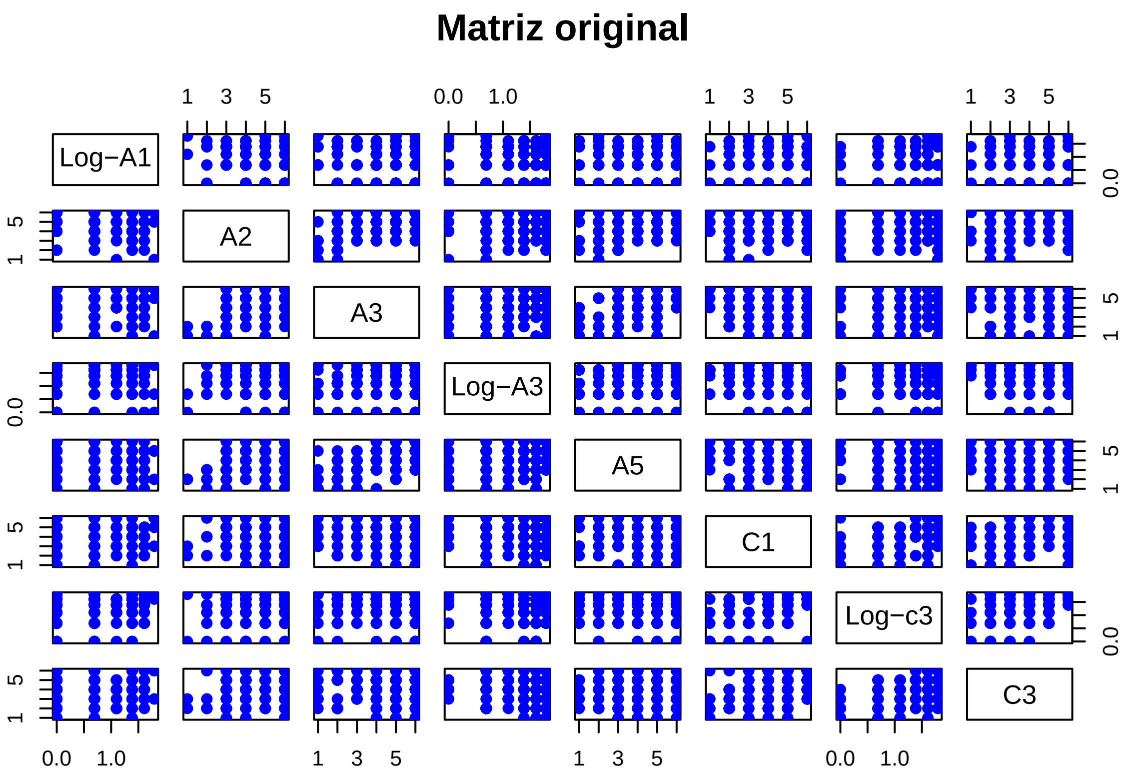
```
x[,1]<-log(x[,1])
colnames(x)[1]<-"Log-A1"

x[,4]<-log(x[,4])
colnames(x)[4]<-"Log-A3"

x[,7]<-log(x[,7])
colnames(x)[7]<-"Log-c3"
```

Grafico scater para la visualizacion de la matriz original con 3 variables que se incluyeron.

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



se va a implementar la matriz de correlaciones para estimar la matriz de carga

Reduccion de la dimensionalidad

#Análsis Factorial de componentes principales (PCFA)

1.- Calcular la matriz de medias y de correlaciones

Matriz de medias

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

```
##    Log-A1        A2        A3    Log-A3        A5        C1    Log-c3        C3
## 0.6421058 4.7989691 4.5876289 1.3758019 4.3608247 4.3711340 1.3523109 4.1907216
```

Matriz de correlaciones

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

```
##             Log-A1         A2          A3      Log-A3          A5          C1
## Log-A1  1.00000000 -0.4423689 -0.28929840 -0.03864516 -0.28173328 -0.12424303
## A2     -0.44236890  1.0000000  0.56013797  0.21712125  0.44409332  0.18961305
## A3     -0.28929840  0.5601380  1.00000000  0.29120940  0.61294435  0.15483591
## Log-A3 -0.03864516  0.2171213  0.29120940  1.00000000  0.29504031  0.06777130
## A5     -0.28173328  0.4440933  0.61294435  0.29504031  1.00000000  0.07112259
## C1     -0.12424303  0.1896130  0.15483591  0.06777130  0.07112259  1.00000000
## Log-c3 -0.04504144  0.1813809  0.10839483  0.13783350  0.03285852  0.49153682
## C3     -0.03238919  0.1804103  0.06681774  0.07504765  0.07682045  0.43140371
##             Log-c3          C3
## Log-A1 -0.04504144 -0.03238919
## A2      0.18138087  0.18041028
## A3      0.10839483  0.06681774
```

```
## Log-A3  0.13783350  0.07504765
## A5      0.03285852  0.07682045
## C1      0.49153682  0.43140371
## Log-c3  1.00000000  0.49342431
## C3      0.49342431  1.00000000
```

2.- Reducción de la dimensionalidad mediante

Análisis factorial de componentes principales (PCFA).

1.- Calcular los valores y vectores propios.

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

2.- Valores propios

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

```
## [1] 2.6911802 1.7443655 1.0079287 0.6835086 0.5760598 0.5064023 0.4577060
## [8] 0.3328490
```

3.- Vectores propios

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

```
##            [,1]        [,2]        [,3]        [,4]        [,5]       [,6]
## [1,]  0.3100895 -0.19313432  0.61542451  0.56000888 -0.06472600 -0.2879037
## [2,] -0.4670800  0.16288579 -0.20762404 -0.03609929  0.19677703 -0.5388955
## [3,] -0.4592510  0.27220431  0.09310913  0.35091774 -0.20062410 -0.1906465
## [4,] -0.2647912  0.09641253  0.72694162 -0.60434872  0.01347816  0.1069599
## [5,] -0.4165121  0.31204832  0.15105523  0.39694238  0.02746644  0.4319112
## [6,] -0.2927454 -0.47170392 -0.11452834  0.01333689 -0.68334976  0.2862610
## [7,] -0.2785612 -0.52504129  0.06998016 -0.09134389 -0.03287823 -0.4535500
## [8,] -0.2605309 -0.50717817  0.01413687  0.17498977  0.66921997  0.3235990
##             [,7]        [,8]
## [1,]  0.23120310 -0.18344881
## [2,]  0.45476451 -0.41827411
## [3,]  0.01653452  0.71155550
## [4,]  0.12112985  0.02478375
## [5,] -0.37290863 -0.47159307
## [6,]  0.33798579 -0.12391237
## [7,] -0.65058304 -0.05871559
## [8,]  0.21962469  0.20800485
```

4.- Calcular la proporcion de variabilidad

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

```
## [1] 0.33639752 0.21804568 0.12599108 0.08543857 0.07200747 0.06330029 0.05721325
## [8] 0.04160613
```

5.- Calcular la proporcion de variabilidad acumulada

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

```
## [1] 0.3363975 0.5544432 0.6804343 0.7658729 0.8378803 0.9011806 0.9583939
## [8] 1.0000000
```

Estimacion de la matriz de carga

Nota: se estima la matriz de carga usando los autovalores y autovectores. se aplica la rotación varimax

Primera estimación de Lamda mayuscula se calcula multiplicando la matriz de los 3 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,]  0.5086962 -0.2550811  0.61785944
## [2,] -0.7662362  0.2151305 -0.20844551
## [3,] -0.7533929  0.3595123  0.09347751
## [4,] -0.4343851  0.1273363  0.72981777
## [5,] -0.6832805  0.4121360  0.15165288
## [6,] -0.4802435 -0.6230003 -0.11498147
## [7,] -0.4569745 -0.6934453  0.07025704
## [8,] -0.4273961 -0.6698527  0.01419280
```

Rotación varimax

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

```
## $loadings
##
## Loadings:
##      [,1]   [,2]   [,3]
## [1,]  0.772         0.329
## [2,] -0.790 -0.189  0.132
## [3,] -0.720         0.429
## [4,] -0.101         0.849
## [5,] -0.665         0.466
## [6,] -0.147 -0.780
## [7,]        -0.826  0.106
## [8,]        -0.793
##
##                 [,1]  [,2]  [,3]
## SS loadings    2.214 1.968 1.262
## Proportion Var 0.277 0.246 0.158
## Cumulative Var 0.277 0.523 0.680
##
## $rotmat
##            [,1]       [,2]       [,3]
## [1,]  0.7920418 0.48637686 -0.3689272
## [2,] -0.4510840 0.87348803  0.1831444
## [3,]  0.4113307 0.02135914  0.9112359
```

#Estimación de la matriz de los errores 1.- Estimación de 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,] 0.2944116 0.0000000 0.000000 0.0000000 0.000000 0.0000000 0.0000000
## [2,] 0.0000000 0.3231515 0.000000 0.0000000 0.000000 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000 0.294412 0.0000000 0.000000 0.0000000 0.0000000
```

```
## [4,] 0.0000000 0.0000000 0.000000 0.2624611 0.000000 0.0000000 0.0000000
## [5,] 0.0000000 0.0000000 0.000000 0.0000000 0.340273 0.0000000 0.0000000
## [6,] 0.0000000 0.0000000 0.000000 0.0000000 0.000000 0.3680162 0.0000000
## [7,] 0.0000000 0.0000000 0.000000 0.0000000 0.000000 0.0000000 0.3053719
## [8,] 0.0000000 0.0000000 0.000000 0.0000000 0.000000 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,] 0.3684285
```

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

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

```
##              Log-A1          A2          A3      Log-A3          A5          C1
## Log-A1  0.70558843 -0.4423689 -0.28929840 -0.03864516 -0.28173328 -0.12424303
## A2     -0.44236890  0.6768485  0.56013797  0.21712125  0.44409332  0.18961305
## A3     -0.28929840  0.5601380  0.70558804  0.29120940  0.61294435  0.15483591
## Log-A3 -0.03864516  0.2171213  0.29120940  0.73753890  0.29504031  0.06777130
## A5     -0.28173328  0.4440933  0.61294435  0.29504031  0.65972699  0.07112259
## C1     -0.12424303  0.1896130  0.15483591  0.06777130  0.07112259  0.63198384
## Log-c3 -0.04504144  0.1813809  0.10839483  0.13783350  0.03285852  0.49153682
## C3     -0.03238919  0.1804103  0.06681774  0.07504765  0.07682045  0.43140371
##              Log-c3          C3
## Log-A1 -0.04504144 -0.03238919
## A2      0.18138087  0.18041028
## A3      0.10839483  0.06681774
## Log-A3  0.13783350  0.07504765
## A5      0.03285852  0.07682045
## C1      0.49153682  0.43140371
## Log-c3  0.69462813  0.49342431
## C3      0.49342431  0.63157145
```

Calculo de la matriz de autovalores y autovectores

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

Autovalores

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

```
## [1] 2.37312991 1.40700855 0.72814231 0.39183190 0.21362634 0.17803340 0.13410617
## [8] 0.01759573
```

Autovectores

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

```
## [1] 2.37312991 1.40700855 0.72814231 0.39183190 0.21362634 0.17803340 0.13410617
## [8] 0.01759573
```

Proporcion de variabilidad

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

```
## [1] 0.435958687 0.258476199 0.133764259 0.071981952 0.039244484 0.032705841
## [7] 0.024636135 0.003232444
```

Proporcion de variabilidad acumulada

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

```
## [1] 0.4359587 0.6944349 0.8281991 0.9001811 0.9394256 0.9721314 0.9967676
## [8] 1.0000000
```

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,]  0.4846316 -0.2283072 -0.5060177168
## [2,] -0.7198201  0.1815417  0.1708825503
## [3,] -0.7179006  0.3186534 -0.0647604825
## [4,] -0.4216490  0.1119868 -0.6458561865
## [5,] -0.6391704  0.3571606 -0.1041350219
## [6,] -0.4348544 -0.5544242  0.0899633019
## [7,] -0.4249892 -0.6473064 -0.0512407567
## [8,] -0.3856151 -0.5948869 -0.0005312241
```

Rotacion varimax

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

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,] 0.4569541 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000
## [2,] 0.0000000 0.4197007 0.0000000 0.0000000 0.0000000 0.0000000 0.000000
## [3,] 0.0000000 0.0000000 0.3788847 0.0000000 0.0000000 0.0000000 0.000000
## [4,] 0.0000000 0.0000000 0.0000000 0.3925409 0.0000000 0.0000000 0.000000
## [5,] 0.0000000 0.0000000 0.0000000 0.0000000 0.4530534 0.0000000 0.000000
## [6,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.4954221 0.000000
## [7,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.397753
## [8,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000
##           [,8]
## [1,] 0.0000000
## [2,] 0.0000000
## [3,] 0.0000000
## [4,] 0.0000000
## [5,] 0.0000000
## [6,] 0.0000000
## [7,] 0.0000000
## [8,] 0.4974103
```

Obtencion de los scores de ambos métodos

PCFA

```
FS.est.1<-scale(x)%*% as.matrix(L.est.1.var$loadings)
FS.est.1[1:10,]
```

```
##              [,1]        [,2]       [,3]
## 61617  1.9791718  2.88209585 -0.7180140
## 61618  0.1464406 -0.07437579 -0.9706598
## 61620  1.8160496  0.02656501  0.5484395
## 61621  0.4563440  1.01352840  1.4160369
## 61622  1.8677855  0.02489783 -0.4003399
## 61623 -0.1897020 -3.18346375  2.0276925
## 61624 -0.6261198 -0.31205111 -0.1151509
## 61629  6.0136164  2.58215998 -1.8352571
## 61630  2.0047552 -0.78196873 -0.3022537
## 61633 -1.5006909 -2.78157306  1.5760312
```

PFA

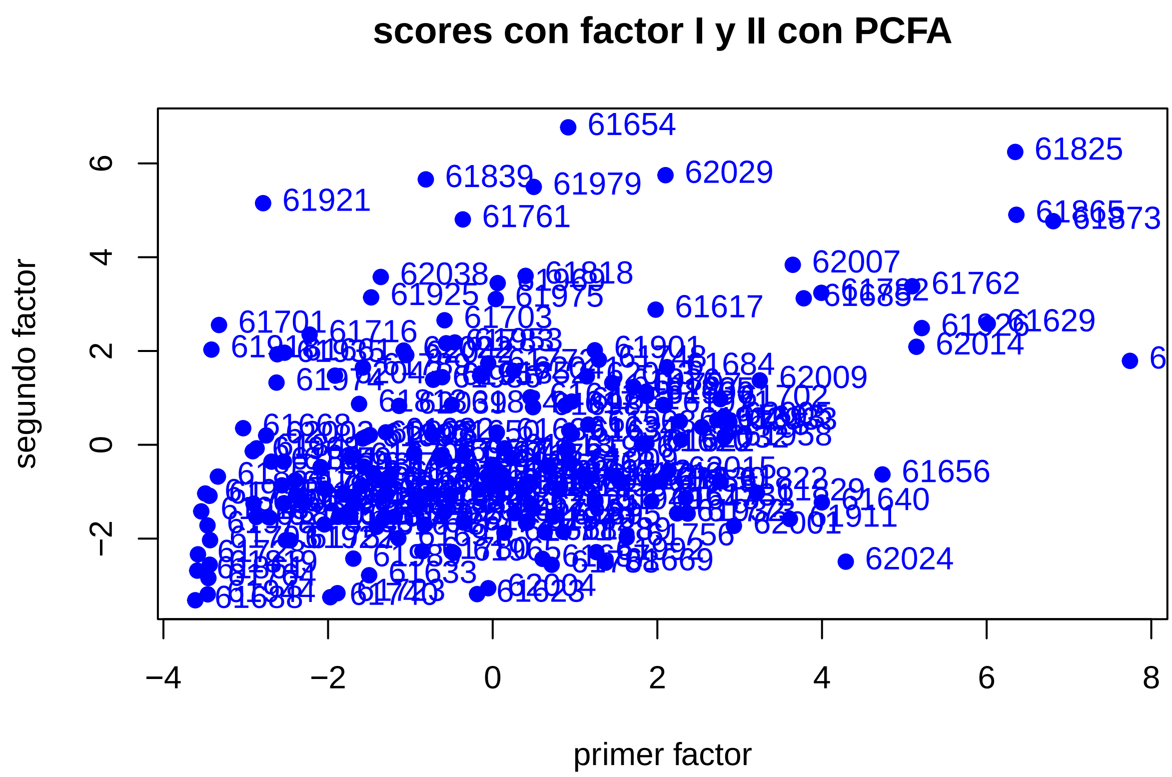
```
FS.est.2<-scale(x)%*% as.matrix (L.est.2.var$loadings)
FS.est.2[1:10,]
```

```
##              [,1]        [,2]       [,3]
## 61617  1.8620289  2.63257005  0.7052977
## 61618  0.1551565 -0.05474727  0.9006988
## 61620  1.6365418  0.04040420 -0.3772022
## 61621  0.3671053  0.90276745 -1.1644564
## 61622  1.7235916  0.06695653  0.4251292
## 61623 -0.2589965 -2.89078635 -1.7600629
## 61624 -0.5736358 -0.29256800  0.1044518
## 61629  5.6012917  2.48825261  1.7324862
## 61630  1.8263835 -0.67867446  0.3400254
## 61633 -1.4542890 -2.53760825 -1.4365721
```

Graficamos los scores

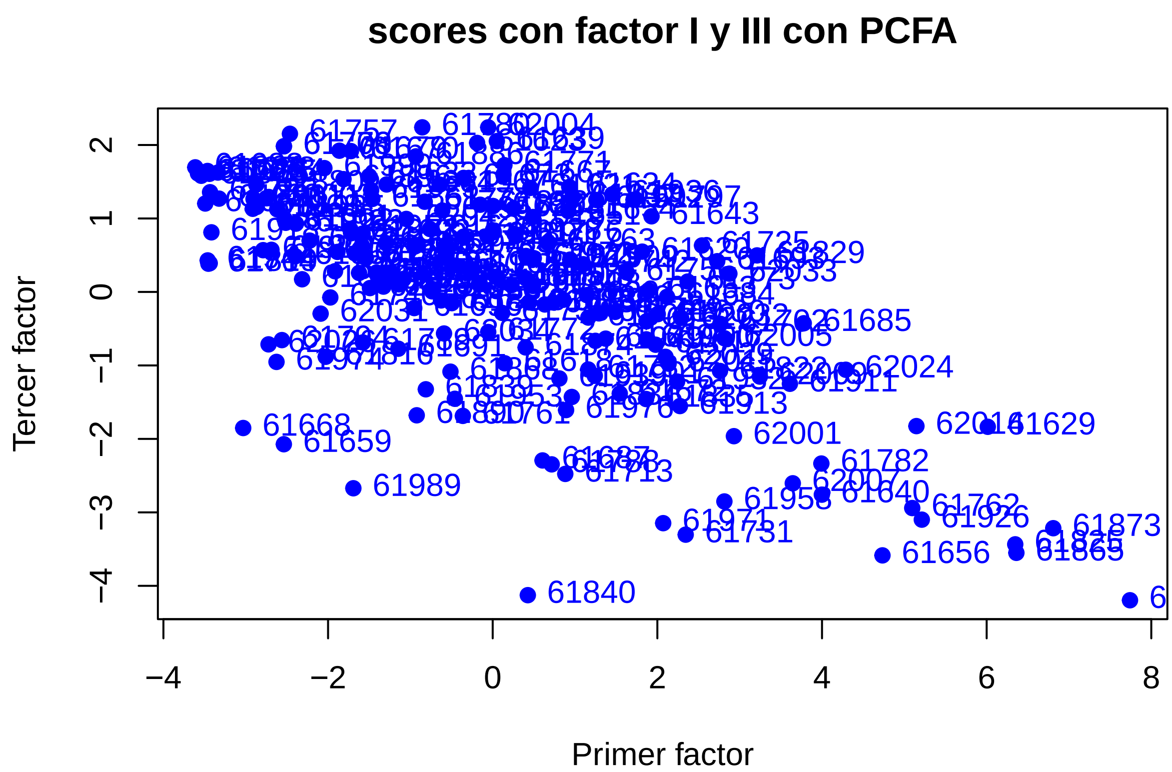
Factor I y II

```
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="blue")
```



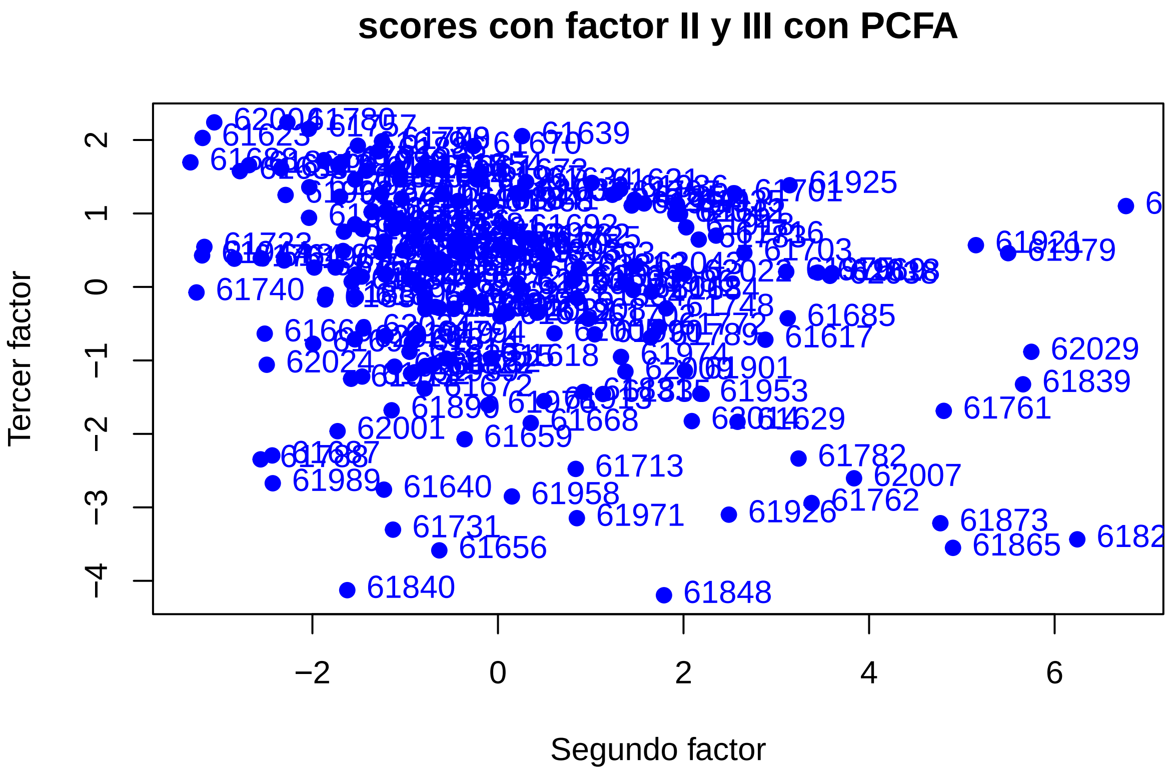
Factor I y III

```
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="blue")
```



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="blue")
text(FS.est.1[,2], FS.est.1[,3], labels = rownames(x), pos=4, col="blue")
```


scores con factor II y III con PCFA
Tercer factor
Segundo factor
2
1
0
−1
−2
−3
−4
−2
0
2
4
6
61639
61925
61921
61979
61740
61685
61617
62029
61839
61761
62024
62001
61659
61668
61629
61713
61782
62007
61989
61640
61958
61971
61926
61762
61873
61865
61731
61656
61840
61848
61953
61901
61974
61618
61890
61670
61623
61722