Análisis Factorial

Saraí Campos Varela

2022-05-24

Análisis Factorial

Introducción

El análisis factorial es un método de reducción estadística que tiene como objetivo explicar las posibles correlaciones entre ciertas variables. Para ello, teniendo en cuenta el efecto de otras, los factores, que no son observables.

Matriz de trabajo

Se trabajo con la matriz statex77, extraída del paquete datos que se encuentra precargado en R, es una matriz de datos cuantitativos y contiene informacionde los de EU.

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

Quitar los espacios de los nombres.

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

Separa n (estados) y p (variables).

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

Exploración de la matriz.

Dimensión de la matriz. La matriz cuenta con 50 observaciones y 8 variables.

```
dim(x)
```

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

Tipo de variables.

```r
str(x)
```

```
## 'data.frame':	50 obs. of  8 variables:
##  $ Population: num  3615 365 2212 2110 21198 ...
##  $ Income    : num  3624 6315 4530 3378 5114 ...
##  $ Illiteracy: num  2.1 1.5 1.8 1.9 1.1 0.7 1.1 0.9 1.3 2 ...
##  $ Life.Exp  : num  69 69.3 70.5 70.7 71.7 ...
##  $ Murder    : num  15.1 11.3 7.8 10.1 10.3 6.8 3.1 6.2 10.7 13.9 ...
##  $ HS.Grad   : num  41.3 66.7 58.1 39.9 62.6 63.9 56 54.6 52.6 40.6 ...
##  $ Frost     : num  20 152 15 65 20 166 139 103 11 60 ...
##  $ Area      : num  50708 566432 113417 51945 156361 ...
```

Como se mencionó, la matriz de datos es cuantitativa.

Nombre de las variables.

```r
colnames(x)
```

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

Se buscan datos perdidos en la matriz.

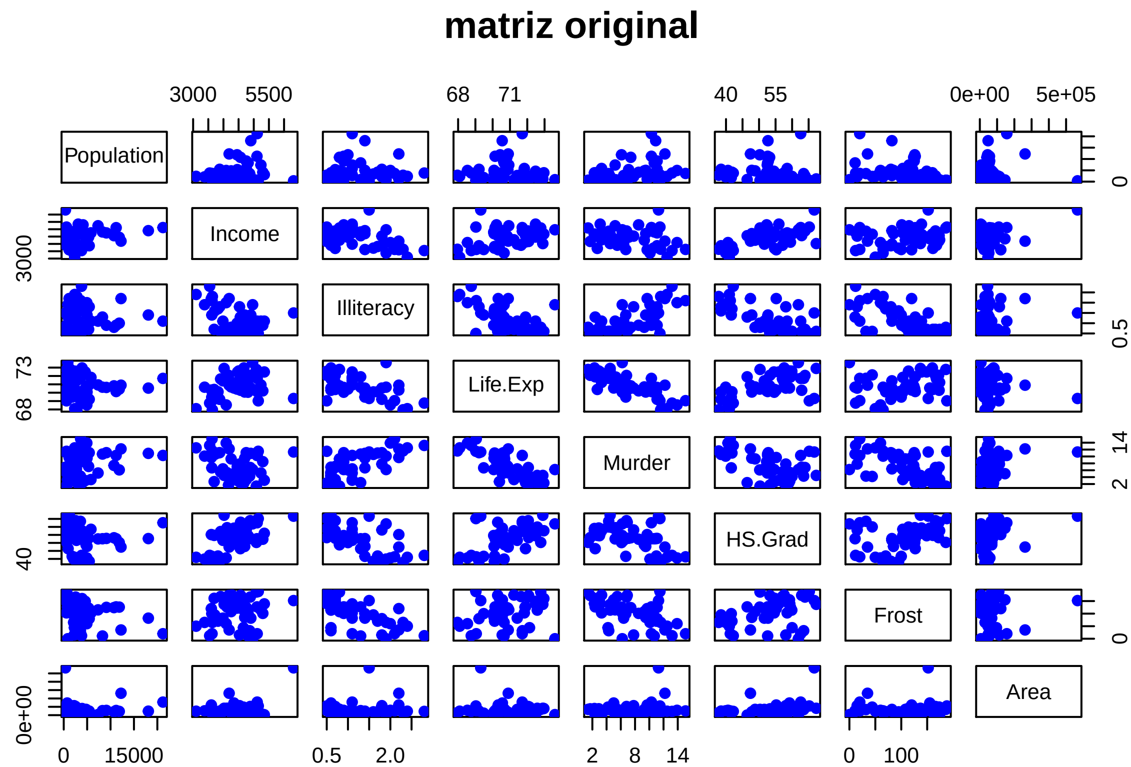
```r
anyNA(x)
```

```
## [1] FALSE
```

No se encuentran valores nulos en la matriz.

Generación de un scater plot para la visualización de variables originales.

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



Transformación de alguna variables.

Aplicamos logaritmo para las columnas 1,3 y 8

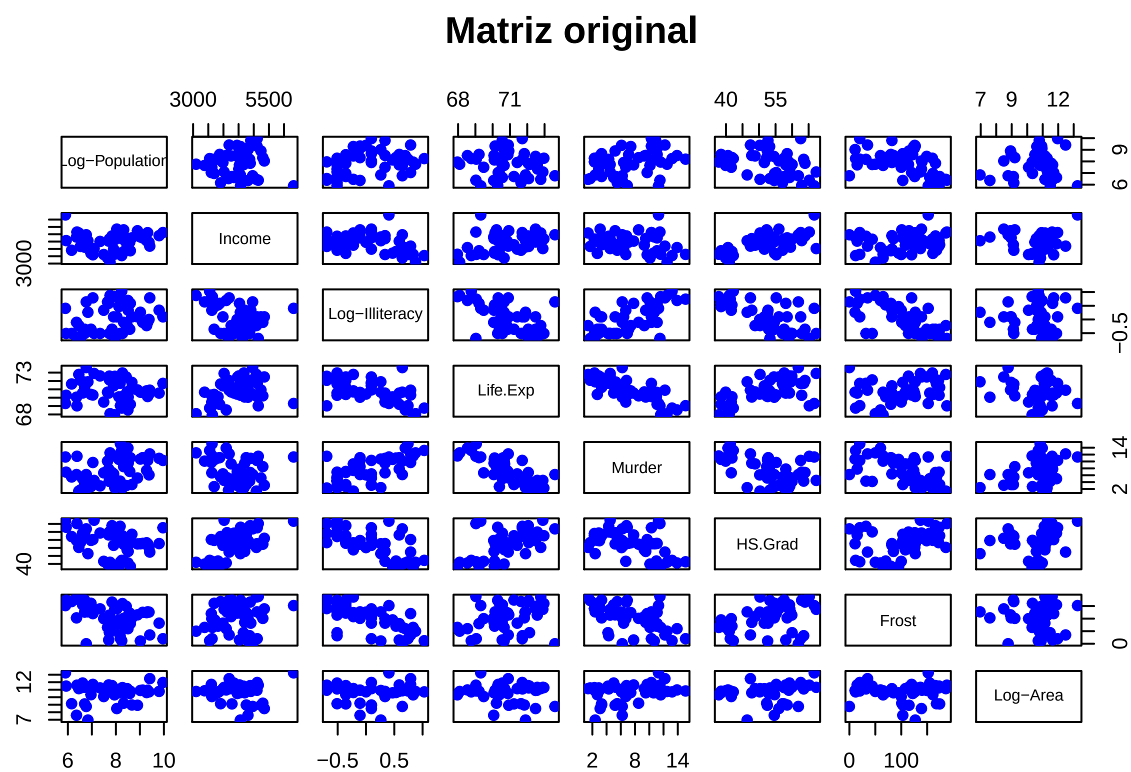
```
x[,1]<-log(x[,1])
colnames(x)[1]<-"Log-Population"

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

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

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

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



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

Análsis Factorial de componentes principales (PCFA)

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
```

Calcular los valores y vectores propios.

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

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

```
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
```

Estimacion de la matriz de carga

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

Se hace la primera estimación de Lamda mayúscula y se calcula multiplicando la matriz de los 3 primeros autovectores por la matriz diagonal formada por la raíz 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
```

Estimación de la matriz de los errores

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,] 0.2871756 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.

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

Autovalores

```r
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

```r
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

```r
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
```

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)
```

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)
FS.est.1
```

```
##                       [,1]          [,2]       [,3]
## Alabama        -5.84072356 -1.3993671511  4.0008109
## Alaska          2.12443806 -3.6163397014 -1.3435941
## Arizona        -0.77245459 -1.1030150088  1.7864181
## Arkansas       -4.26961555 -0.1287634469  1.8680205
## California      1.57843978 -1.6386262821  3.0959757
## Colorado        3.35619481 -0.5747409714 -1.9955520
## Connecticut     2.96609993  2.5265114588 -1.0120520
## Delaware        0.15111765  2.2707877284 -1.3473631
## Florida        -0.91278118 -0.8518787165  3.2141818
## Georgia        -5.10406769 -1.5374188978  3.5972606
## Hawaii          1.68679592  2.0782245763  0.6972161
## Idaho           1.93931571  0.0374520725 -2.6403015
## Illinois        0.36572803 -0.9730363911  1.3246992
## Indiana         0.69870165  0.1740586327 -0.1660034
## Iowa            3.77325852  0.8634090197 -2.4308546
## Kansas          3.22079390  0.2206198504 -1.7333568
## Kentucky       -3.97957229 -0.1711842990  1.8581455
## Louisiana      -6.15095874 -1.1449716511  4.2193388
## Maine           0.38912287  0.9352663421 -2.8385772
## Maryland        0.54556931  0.6481615589  0.7313943
## Massachusetts   1.95531363  1.9508870989 -0.0699601
## Michigan        0.06109118 -0.8995742724  1.1610156
## Minnesota       3.83625590  0.7199310360 -2.2609012
## Mississippi    -6.73875213 -1.1336057288  3.0124928
## Missouri       -0.63621057 -0.5673516660  0.5606479
## Montana         1.70022911 -0.7530855537 -2.9827203
## Nebraska        3.31393569  0.5702899251 -2.6630094
## Nevada          1.83953234 -2.1624547546 -2.8632403
## New Hampshire   1.76672303  1.8835104424 -3.2522623
## New Jersey      1.23076573  1.5154423999  0.6483326
## New Mexico     -2.42369795 -1.2184859435  0.1095350
## New York       -0.55160991 -0.8431042602  2.9025469
## North Carolina -4.53932589 -0.7126552652  2.8168209
## North Dakota    3.26810535  1.0664889529 -3.5180166
## Ohio            0.67643704 -0.0394642439  0.5816740
## Oklahoma       -0.43628926  0.0293430043  0.2108486
## Oregon          2.64633236 -0.0126633017 -0.6563722
```

```
## Pennsylvania   -0.06313819  0.0425262164  0.8538298
## Rhode Island    0.25059508  4.0533333045 -1.3779994
## South Carolina -6.20030464 -0.7067780563  3.0142562
## South Dakota    2.51505516  0.8539599931 -3.9694575
## Tennessee      -3.75602365 -0.3764569265  2.4225536
## Texas          -2.74825842 -2.0176142597  4.0126966
## Utah            3.40911641  0.2638533973 -3.0642167
## Vermont         1.26368503  1.7670538099 -3.5748058
## Virginia       -1.45435214 -0.4332714574  1.8388594
## Washington      2.95298764  0.0002978623 -0.1436737
## West Virginia  -3.41599674  0.5649932020  0.5132111
## Wisconsin       2.58972274  0.8701285803 -1.5397225
## Wyoming         1.92267355 -0.8906222579 -3.6087703
```

PFA

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

```
##                       [,1]         [,2]       [,3]
## Alabama        -5.69766092 -1.133005866  3.9030908
## Alaska          1.77921500 -3.310049553 -1.2425530
## Arizona        -0.80948635 -1.007423566  1.6833688
## Arkansas       -4.04451164 -0.036340306  1.8899610
## California      1.28900772 -1.589528660  2.7938220
## Colorado        3.21256763 -0.645092519 -1.9103448
## Connecticut     2.85639977  2.291700954 -1.1152442
## Delaware        0.22491218  2.168332191 -1.3109174
## Florida        -1.04778981 -0.760012075  2.9630979
## Georgia        -5.04193484 -1.243399542  3.4848855
## Hawaii          1.64548810  1.848120424  0.5487863
## Idaho           1.99602286 -0.067186945 -2.4442739
## Illinois        0.17329771 -0.870927790  1.1838509
## Indiana         0.66348403  0.140717116 -0.1900850
## Iowa            3.70915552  0.657976435 -2.3698485
## Kansas          3.13617617  0.071725764 -1.6894853
## Kentucky       -3.82119443 -0.051170443  1.8492550
## Louisiana      -5.97309240 -0.880509145  4.1021292
## Maine           0.58567717  0.845398887 -2.6098620
## Maryland        0.40855637  0.650876372  0.5867974
## Massachusetts   1.91021424  1.761365924 -0.1964750
## Michigan       -0.07208772 -0.823049544  1.0671998
## Minnesota       3.74953682  0.518054623 -2.2104937
## Mississippi    -6.45121865 -0.852611917  3.0320154
## Missouri       -0.64446964 -0.519762510  0.5472506
## Montana         1.72574501 -0.752576236 -2.7507980
## Nebraska        3.28773039  0.392513546 -2.5439122
## Nevada          1.69672312 -1.994626548 -2.6292009
## New Hampshire   1.87991014  1.704867403 -3.0632652
## New Jersey      1.10782292  1.425042094  0.4638907
## New Mexico     -2.26112419 -1.086582245  0.2653217
## New York       -0.72255151 -0.744949928  2.6624378
```

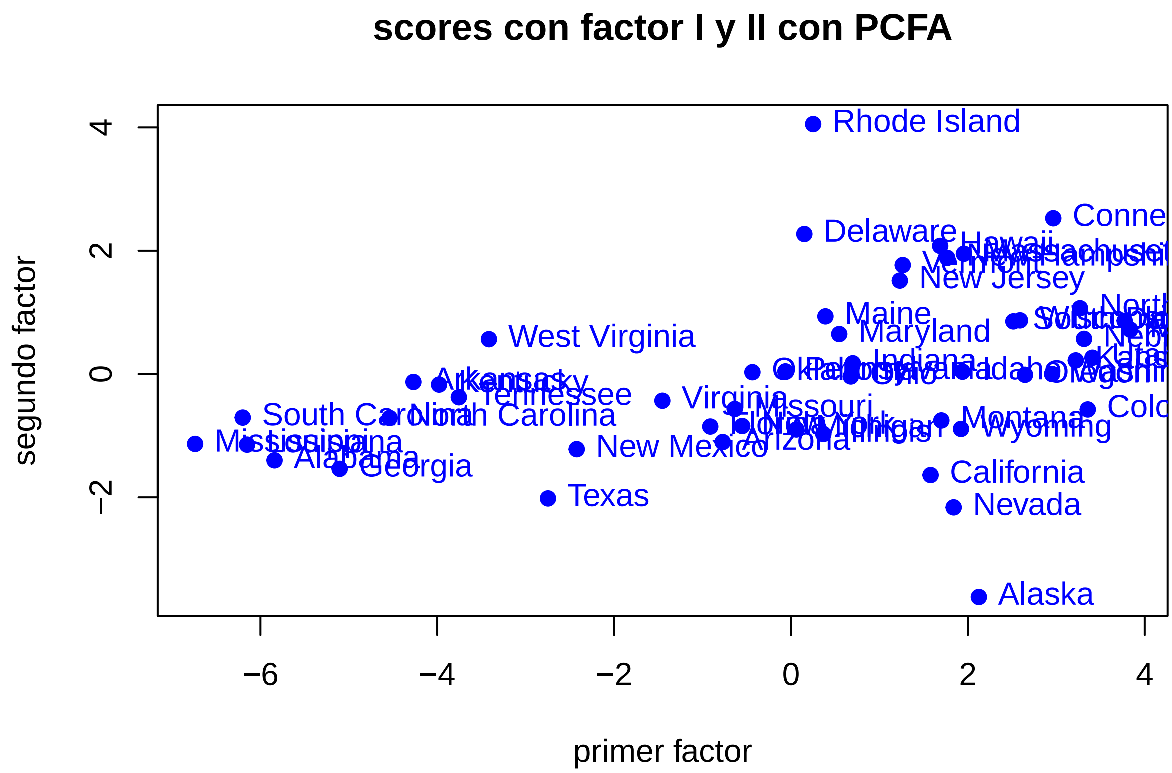
```
## North Carolina -4.42441540 -0.513264749  2.7372284
## North Dakota    3.22068093  0.897031063 -3.3556310
## Ohio            0.59453054 -0.051780182  0.4905274
## Oklahoma       -0.36512462  0.000708499  0.2244101
## Oregon          2.56050584 -0.129810062 -0.6934180
## Pennsylvania   -0.10451900  0.054229408  0.7553645
## Rhode Island    0.40356926  3.785456289 -1.3760426
## South Carolina -5.98815271 -0.435831413  2.9745853
## South Dakota    2.60764548  0.683975660 -3.7117087
## Tennessee      -3.63769564 -0.249263663  2.3593673
## Texas          -2.80670233 -1.827474308  3.8156526
## Utah            3.44131011  0.069209103 -2.8669774
## Vermont         1.44160727  1.580578146 -3.3086066
## Virginia       -1.50774364 -0.328200587  1.7151967
## Washington      2.81601549 -0.109025242 -0.2503494
## West Virginia  -3.18525955  0.632647668  0.5745805
## Wisconsin       2.55487697  0.699000994 -1.5141208
## Wyoming         1.92835024 -0.866073018 -3.3204601
```

Graficamos ambos scores

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

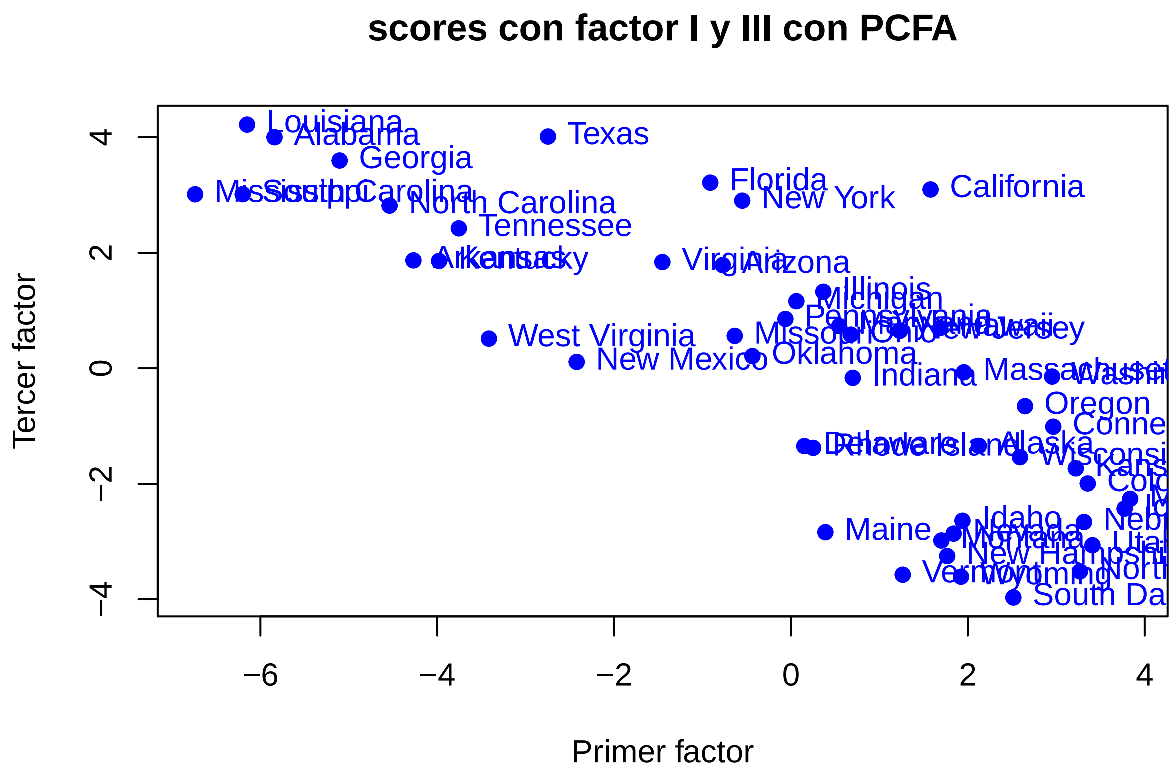
Factor I y II

```r
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")
```



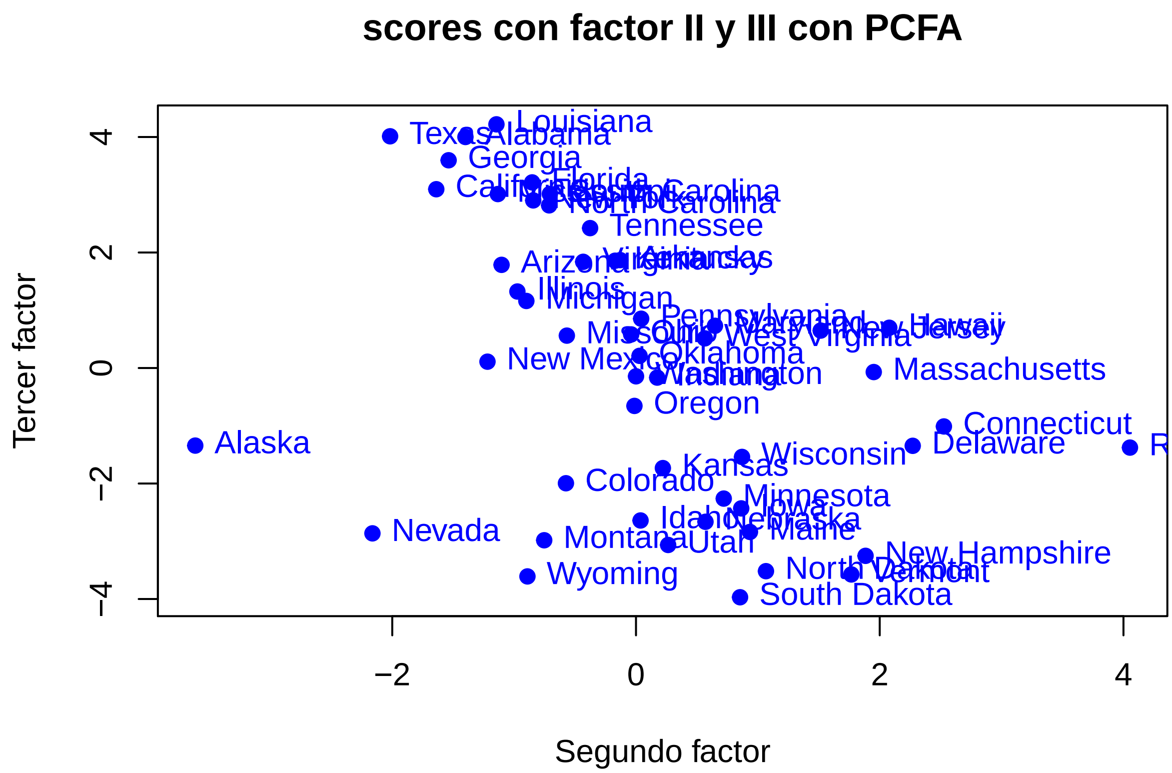
Factor I y III

```
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")
```



PRACTICA PSICOLOGÍA

Librerias

```
library(psych)
library(polycor)
library(ggcorrplot)
```

Extraccion de datos

Se encuentra dentro de la paquetería *psych*

```
x = bfi
```

Exploracion de la matriz

Dimension de la matriz

```
dim(x)
```

```
## [1] 2800   28
```

Tipos de variables

```
str(x)
```

```
## 'data.frame':    2800 obs. of  28 variables:
##  $ A1       : int  2 2 5 4 2 6 2 4 4 2 ...
##  $ A2       : int  4 4 4 4 3 6 5 3 3 5 ...
##  $ A3       : int  3 5 5 6 3 5 5 1 6 6 ...
##  $ A4       : int  4 2 4 5 4 6 3 5 3 6 ...
##  $ A5       : int  4 5 4 5 5 5 5 1 3 5 ...
##  $ C1       : int  2 5 4 4 4 6 5 3 6 6 ...
##  $ C2       : int  3 4 5 4 4 6 4 2 6 5 ...
##  $ C3       : int  3 4 4 3 5 6 4 4 3 6 ...
##  $ C4       : int  4 3 2 5 3 1 2 2 4 2 ...
##  $ C5       : int  4 4 5 5 2 3 3 4 5 1 ...
##  $ E1       : int  3 1 2 5 2 2 4 3 5 2 ...
##  $ E2       : int  3 1 4 3 2 1 3 6 3 2 ...
##  $ E3       : int  3 6 4 4 5 6 4 4 NA 4 ...
##  $ E4       : int  4 4 4 4 4 5 5 2 4 5 ...
##  $ E5       : int  4 3 5 4 5 6 5 1 3 5 ...
##  $ N1       : int  3 3 4 2 2 3 1 6 5 5 ...
##  $ N2       : int  4 3 5 5 3 5 2 3 5 5 ...
##  $ N3       : int  2 3 4 2 4 2 2 2 2 5 ...
##  $ N4       : int  2 5 2 4 4 2 1 6 3 2 ...
##  $ N5       : int  3 5 3 1 3 3 1 4 3 4 ...
##  $ O1       : int  3 4 4 3 3 4 5 3 6 5 ...
##  $ O2       : int  6 2 2 3 3 3 2 2 6 1 ...
##  $ O3       : int  3 4 5 4 4 5 5 4 6 5 ...
##  $ O4       : int  4 3 5 3 3 6 6 5 6 5 ...
##  $ O5       : int  3 3 2 5 3 1 1 3 1 2 ...
##  $ gender   : int  1 2 2 2 1 2 1 1 1 2 ...
##  $ education: int  NA NA NA NA NA 3 NA 2 1 NA ...
##  $ age      : int  16 18 17 17 17 21 18 19 19 17 ...
```

int en R denota variables discretas Nombre de las variables

```
colnames(x)
```

```
##  [1] "A1"        "A2"        "A3"        "A4"        "A5"        "C1"
##  [7] "C2"        "C3"        "C4"        "C5"        "E1"        "E2"
## [13] "E3"        "E4"        "E5"        "N1"        "N2"        "N3"
## [19] "N4"        "N5"        "O1"        "O2"        "O3"        "O4"
## [25] "O5"        "gender"    "education" "age"
```

Creación de una nueva base de datos donde se incluyen las variables 1 a 25 y solo usamos 200 observaciones

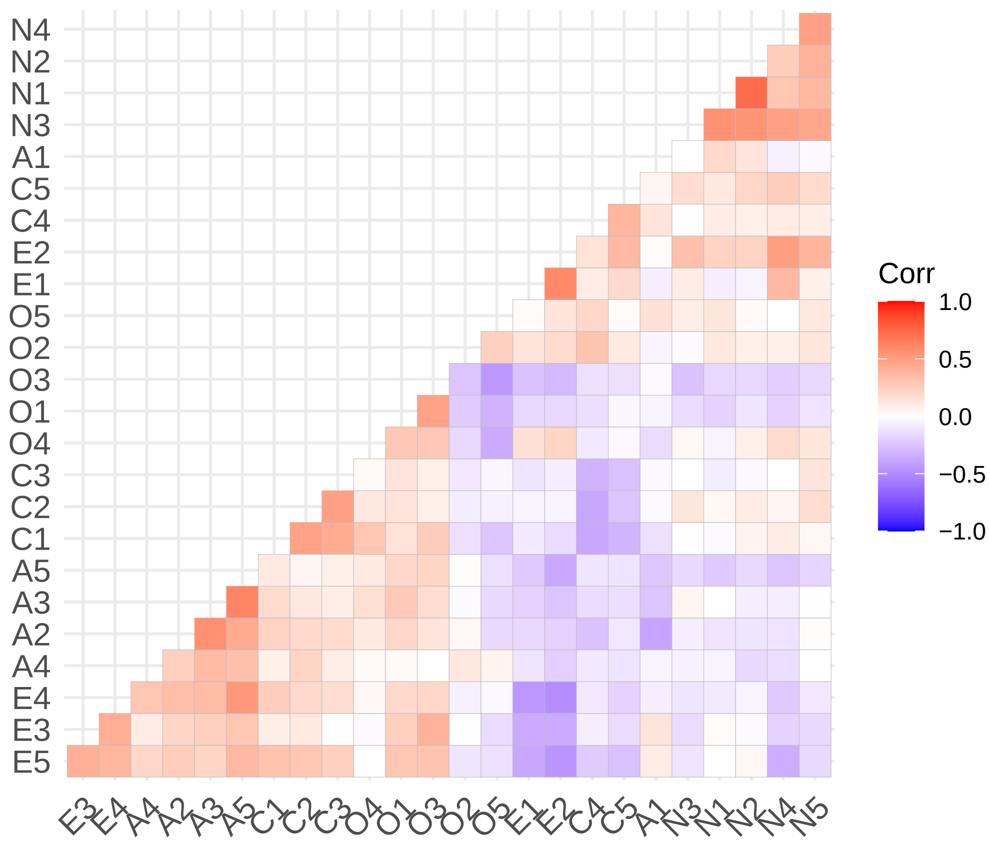
```r
x1= bfi[1:200,1:25]
```

Matriz de correlaciones

```r
R= hetcor(x1)$correlations
```

Gráfico de correlaciones

```r
ggcorrplot(R,type="lower",hc.order= TRUE)
```



Factorización de la matriz de correlaciones

Se utiliza la prueba de esfericidad de Bartlett.

```r
prueba_Bartlett= cortest.bartlett(R)
```

Visualización de el p-valor

```r
prueba_Bartlett$p.value
```

```
## [1] 5.931663e-60
```

H0 : variables correlacionadas H1 : las variables no están correlacionadas

No rechazo H0 con ayuda del p-valor

Criterio Kaiser-Meyer-Olkin

Permite identificar si los datos analizados son adecuados para un análisis factorial.

0.00 a 0.49 No adecuados 0.50 a 0.59 Poco adecuados 0.60 a 0.69 Aceptables 0.70 a 0.89 Buenos 0.90 a 1.00 Excelente

```r
KMO(R)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = R)
## Overall MSA =  0.76
## MSA for each item =
##   A1   A2   A3   A4   A5   C1   C2   C3   C4   C5   E1   E2   E3   E4   E5   N1
## 0.66 0.77 0.69 0.73 0.75 0.74 0.79 0.76 0.76 0.74 0.80 0.81 0.79 0.81 0.83 0.70
##   N2   N3   N4   N5   O1   O2   O3   O4   O5
## 0.67 0.82 0.79 0.82 0.79 0.65 0.81 0.62 0.77
```

Extracción de factores

Modelo varimax

```r
modelo1= fa(R,nfactor=3,rotate = "none",fm = "mle")
```

Modelo dos

```r
modelo2= fa(R,nfactor=3,rotate = "none",fm = "minres")
```

```r
C1= sort(modelo1$communality,decreasing = TRUE)
```

```r
C2= sort(modelo2$communality,decreasing = TRUE)
```

Combinar los resultados para comparar

```r
head(cbind(C1,C2))
```

```
##           C1        C2
## N1 0.7576920 0.6809294
## E2 0.6802809 0.6564523
## N2 0.6797943 0.5866483
## E1 0.5219674 0.5394762
## N3 0.5198285 0.4942059
## N4 0.4839516 0.4744005
```

Extracción de unidades La unicidad es el cuadrado del coeficiente del factor único, y se expresa como la proporción de la varianza explicada por el factor único. es decir, no puede ser explicada por otros factores.

Unicidad del modelo 1

```
u1=sort(modelo1$uniquenesses,decreasing = TRUE)
```

Unicidad del modelo 2

```
u2= sort(modelo2$uniquenesses,decreasing = TRUE)
```

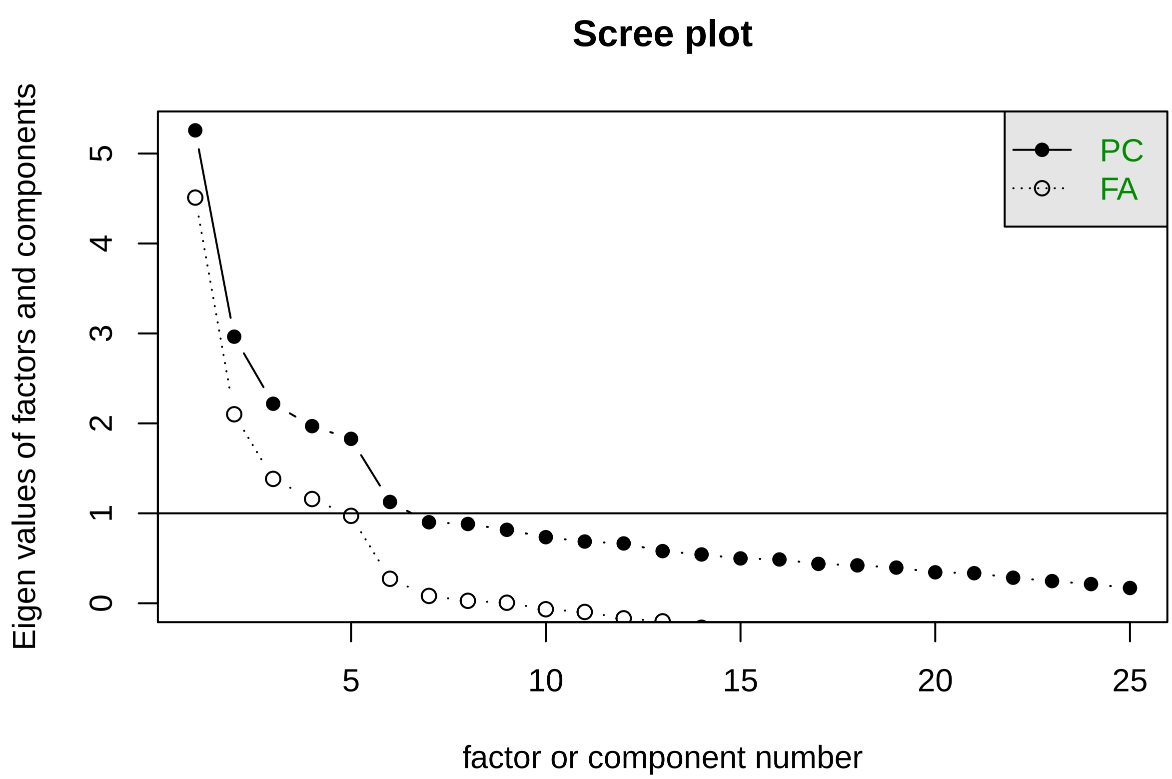
Comparación

```
head(cbind(u1,u2))
```

```
##           u1        u2
## O2 0.9460554 0.9293483
## A4 0.8928892 0.8908844
## A1 0.8607240 0.8822080
## O5 0.8533481 0.8272041
## C5 0.8136600 0.7931685
## O1 0.7986908 0.7904667
```

Elegir el numero de los factores

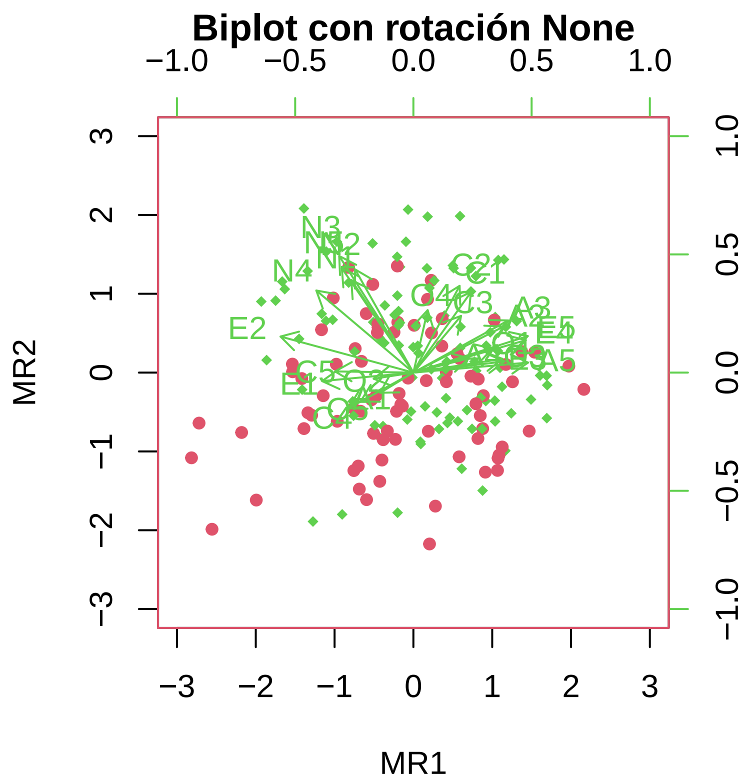
```
scree(R)
```

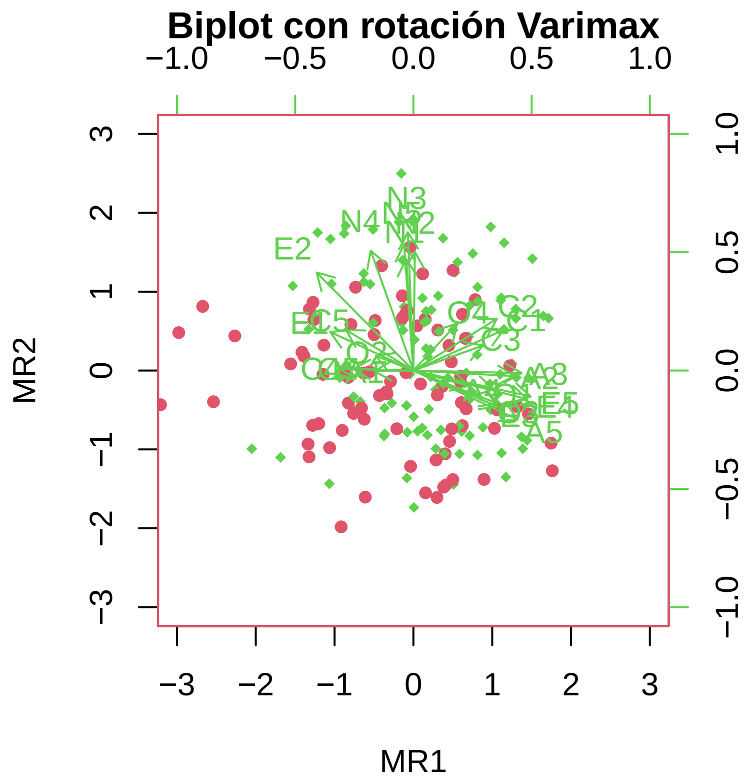


Rotación de la matriz

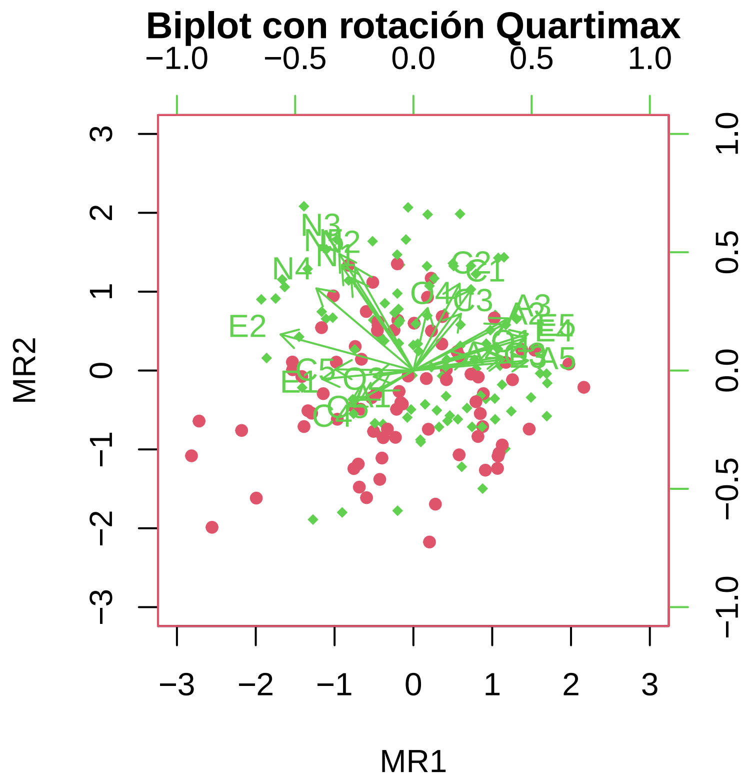
```
library(GPArotation)
```

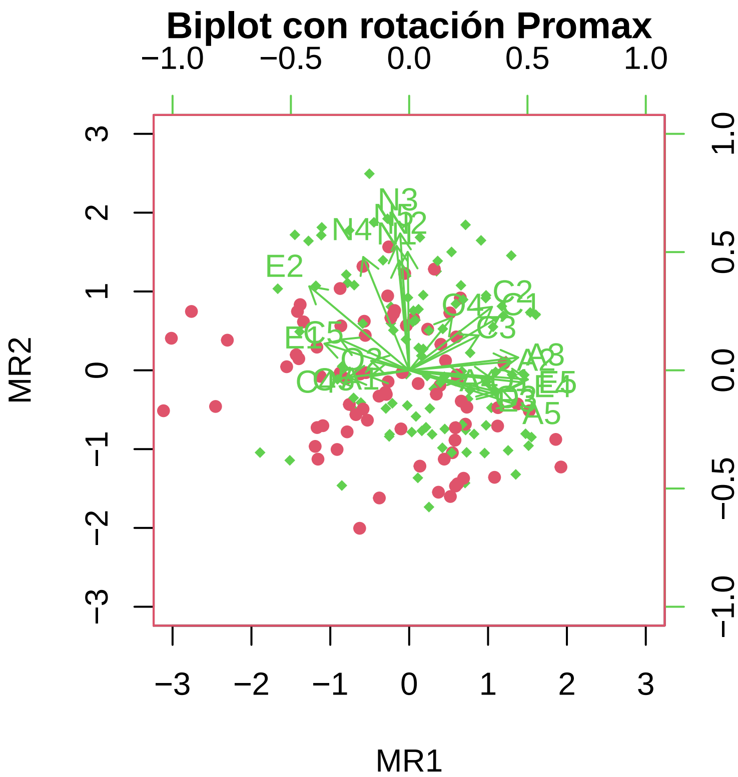
```
rot= c("None", "Varimax", "Quartimax", "Promax")
bi_mod= function(tipo){
  biplot.psych(fa(x1, nfactors = 2,
  fm= "minres", rotate=tipo),
  main = paste("Biplot con rotación", tipo),
  col=c(2,3,4), pch=c(21,18), group=bfi[,"gender"])
}
sapply(rot,bi_mod)
```





Biplot con rotación Varimax
MR1
MR2
N3
N4
N2
N1
E2
E1
C5
C4
C2
C1
C3
A3
E5
E4
A5


Biplot con rotación Quartimax
MR1
MR2
N3
N2
N1
N4
E2
C2
C1
C4
C3
A3
A2
E5
E4
A5
C5
E1
O2
O5
C4

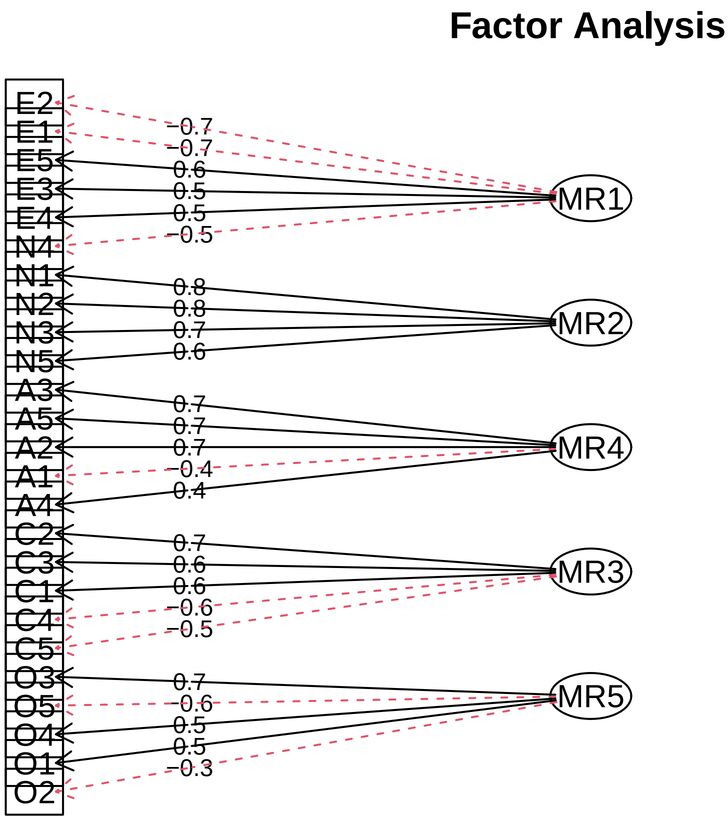



```
## $None
## NULL
##
## $Varimax
## NULL
##
## $Quartimax
## NULL
##
## $Promax
## NULL
```

#interpretación

Creación de un gráfico de árbol

```
modelo_varimax= fa(R,nfactor = 5,
                   rotate = "varimax",
                   fm="minres")
fa.diagram(modelo_varimax)
```



Lineas rojas cargas positivas , lineas negras cargas negativas.

Visualización de la matriz de carga rotada

```
print(modelo_varimax$loadings,cut=0)
```

```
##
## Loadings:
##     MR1    MR2    MR4    MR3    MR5
## A1  0.234  0.106 -0.422 -0.072 -0.092
## A2  0.112 -0.032  0.653  0.190  0.113
## A3  0.198  0.066  0.744  0.051  0.169
## A4  0.163 -0.048  0.413  0.137 -0.142
## A5  0.328 -0.154  0.692 -0.009  0.115
## C1  0.054  0.089  0.140  0.634  0.287
## C2  0.052  0.174  0.114  0.690  0.050
## C3  0.032  0.018  0.076  0.642  0.016
## C4 -0.058  0.087 -0.090 -0.559 -0.159
## C5 -0.241  0.228 -0.040 -0.459  0.014
## E1 -0.691 -0.006 -0.066 -0.084 -0.017
## E2 -0.713  0.345 -0.138 -0.133 -0.025
## E3  0.546  0.003  0.157 -0.008  0.221
## E4  0.522 -0.027  0.416  0.167  0.048
## E5  0.588 -0.009  0.148  0.308  0.159
## N1  0.131  0.802 -0.150 -0.074 -0.133
## N2  0.088  0.800 -0.151 -0.038 -0.008
## N3 -0.183  0.701  0.005  0.037 -0.087
```

```
## N4 -0.513  0.491 -0.006  0.004  0.034
## N5 -0.274  0.571  0.059  0.096 -0.082
## O1  0.203 -0.107  0.148  0.076  0.535
## O2 -0.099  0.096  0.144 -0.191 -0.330
## O3  0.326 -0.159  0.034  0.062  0.680
## O4 -0.240  0.122  0.169  0.105  0.548
## O5 -0.004  0.061 -0.074 -0.077 -0.636
##
##                  MR1   MR2   MR4   MR3   MR5
## SS loadings    2.823 2.667 2.223 2.103 1.867
## Proportion Var 0.113 0.107 0.089 0.084 0.075
## Cumulative Var 0.113 0.220 0.309 0.393 0.467
```