

Análisis Factorial: Autos

Oscar Elí Bonilla Morales

2022-05-22

Selección de datos

```
install.packages("datos")
library(datos)
c <- datos::mtautos
```

Preparación de los datos:

```
autos <- c
head(autos)
```

```
##           millas cilindros cilindrada caballos  eje  peso velocidad
## Mazda RX4      21.0         6         160      110 3.90 2.620      16.46
## Mazda RX4 Wag  21.0         6         160      110 3.90 2.875      17.02
## Datsun 710     22.8         4         108       93 3.85 2.320      18.61
## Hornet 4 Drive  21.4         6         258      110 3.08 3.215      19.44
## Hornet Sportabout 18.7         8         360      175 3.15 3.440      17.02
## Valiant        18.1         6         225      105 2.76 3.460      20.22
##           forma transmision cambios carburadores
## Mazda RX4      0           1         4           4
## Mazda RX4 Wag  0           1         4           4
## Datsun 710     1           1         4           1
## Hornet 4 Drive  1           0         3           1
## Hornet Sportabout 0           0         3           2
## Valiant        1           0         3           1
```

```
str(autos)
```

```
## 'data.frame':  32 obs. of  11 variables:
## $ millas      : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cilindros   : num  6 6 4 6 8 6 8 4 4 6 ...
## $ cilindrada  : num  160 160 108 258 360 ...
## $ caballos    : num  110 110 93 110 175 105 245 62 95 123 ...
## $ eje         : num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ peso        : num  2.62 2.88 2.32 3.21 3.44 ...
## $ velocidad   : num  16.5 17 18.6 19.4 17 ...
## $ forma       : num  0 0 1 1 0 1 0 1 1 1 ...
## $ transmision : num  1 1 1 0 0 0 0 0 0 0 ...
## $ cambios     : num  4 4 4 3 3 3 3 4 4 4 ...
## $ carburadores: num  4 4 1 1 2 1 4 2 2 4 ...
```

Selección de variables

```
library(dplyr)
autos1 <- select(autos, -forma, -transmision) # Forma simple 1
head(autos1)
```

```
##           millas cilindros cilindrada caballos  eje  peso velocidad
## Mazda RX4           21.0         6      160     110 3.90  2.620     16.46
## Mazda RX4 Wag       21.0         6      160     110 3.90  2.875     17.02
## Datsun 710           22.8         4      108      93 3.85  2.320     18.61
## Hornet 4 Drive       21.4         6      258     110 3.08  3.215     19.44
## Hornet Sportabout    18.7         8      360     175 3.15  3.440     17.02
## Valiant              18.1         6      225     105 2.76  3.460     20.22
##           cambios carburadores
## Mazda RX4           4         4
## Mazda RX4 Wag       4         4
## Datsun 710           4         1
## Hornet 4 Drive       3         1
## Hornet Sportabout    3         2
## Valiant              3         1
```

Análisis factorial

Se utilizará la matriz “mtautos”, matriz precargada en el paquete “datos”

Lectura de la matriz de datos

```
x<-as.data.frame(autos1)
head(x)
```

```
##           millas cilindros cilindrada caballos  eje  peso velocidad
## Mazda RX4           21.0         6      160     110 3.90  2.620     16.46
## Mazda RX4 Wag       21.0         6      160     110 3.90  2.875     17.02
## Datsun 710           22.8         4      108      93 3.85  2.320     18.61
## Hornet 4 Drive       21.4         6      258     110 3.08  3.215     19.44
## Hornet Sportabout    18.7         8      360     175 3.15  3.440     17.02
## Valiant              18.1         6      225     105 2.76  3.460     20.22
##           cambios carburadores
## Mazda RX4           4         4
## Mazda RX4 Wag       4         4
## Datsun 710           4         1
## Hornet 4 Drive       3         1
## Hornet Sportabout    3         2
## Valiant              3         1
```

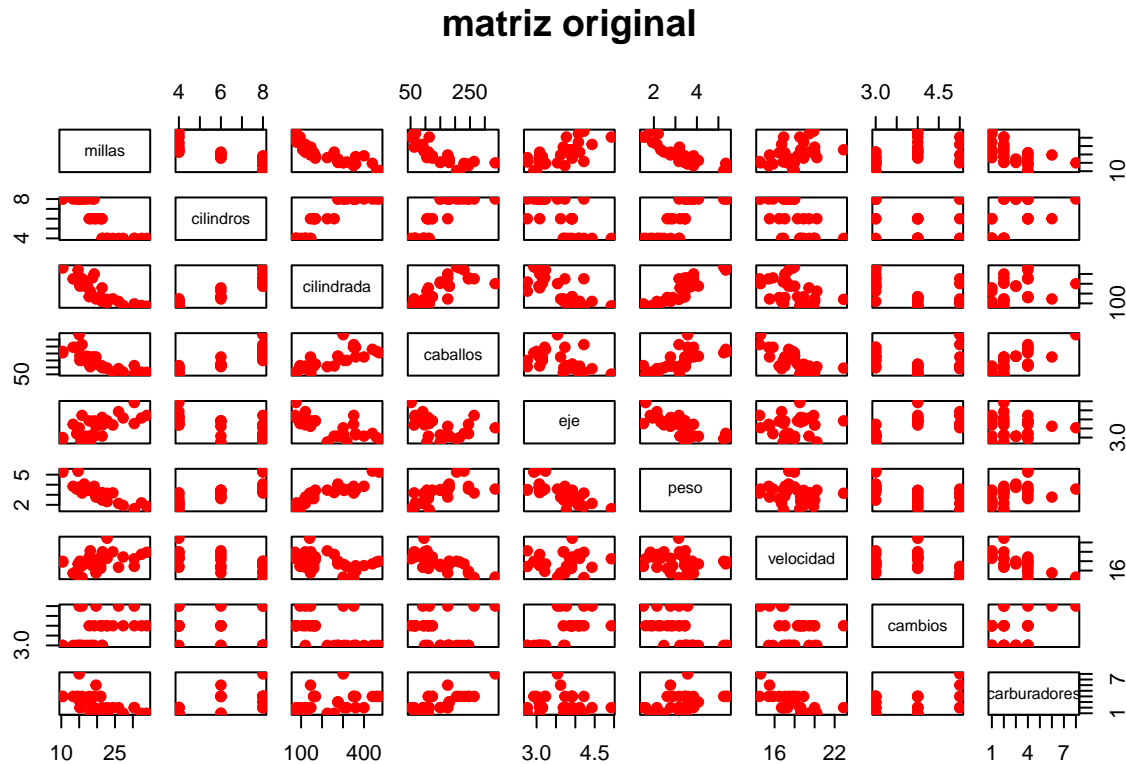
Esta matriz contiene datos sobre los datos las características de distintos automoviles, datos como: “millas”, “caballos”, “peso”, “velocidad”, entre otras.

Separa n (automoviles) y p (variables)

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

Scater plot para la visualización de variables originales

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



En este gráfico se presentan la dispersión y el comportamiento de nuestras variables, en este podemos apreciar algunas.

Transformación de alguna variables

Nota:

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

Reducción de la dimensionalidad

Análisis Factorial de componentes principales (PCFA)

Para obtener el (PCFA) de manera exitosa realizaremos la matriz de medias, al igual que la matriz de correlaciones.

Calcular la matriz de medias y de correlaciones

```
#Matriz de medias
```

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

```
##      millas      cilindros      cilindrada      caballos      eje      peso
## 20.090625    6.187500    230.721875    146.687500    3.596563    3.217250
```

```
##      velocidad      cambios carburadores
##      17.848750      3.687500      2.812500
```

Matriz de correlaciones

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

```
##      millas      cilindros cilindrada      caballos      eje      peso
## millas      1.0000000 -0.8521620 -0.8475514 -0.7761684  0.6811719 -0.8676594
## cilindros  -0.8521620  1.0000000  0.9020329  0.8324475 -0.6999381  0.7824958
## cilindrada -0.8475514  0.9020329  1.0000000  0.7909486 -0.7102139  0.8879799
## caballos   -0.7761684  0.8324475  0.7909486  1.0000000 -0.4487591  0.6587479
## eje        0.6811719 -0.6999381 -0.7102139 -0.4487591  1.0000000 -0.7124406
## peso       -0.8676594  0.7824958  0.8879799  0.6587479 -0.7124406  1.0000000
##      velocidad      cambios carburadores
## millas      0.41868403  0.4802848  -0.5509251
## cilindros  -0.59124207 -0.4926866   0.5269883
## cilindrada -0.43369788 -0.5555692   0.3949769
## caballos   -0.70822339 -0.1257043   0.7498125
## eje        0.09120476  0.6996101  -0.0907898
## peso       -0.17471588 -0.5832870   0.4276059
```

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

Calcular los valores y vectores propios.

Para realizar este paso, se necesitara utilizar los valores originales para calcular los valores propios de nuestra matriz de correlaciones.

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

Valores propios

```
eR$values
```

```
## [1] 5.65593947 2.08210029 0.50421482 0.26502753 0.18315864 0.12379319 0.10506192
## [8] 0.05851375 0.02219038
```

Vectores propios

```
eR$vectors
```

```
##      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.3931477 -0.02753861 -0.22119309 -0.006126378 0.3207620 0.72015586
## [2,] -0.4025537 -0.01570975 -0.25231615 0.040700251 -0.1171397 0.22432550
## [3,] -0.3973528 0.08888469 -0.07825139 0.339493732 0.4867849 -0.01967516
## [4,] -0.3670814 -0.26941371 -0.01721159 0.068300993 0.2947317 0.35394225
## [5,] 0.3118165 -0.34165268 0.14995507 0.845658485 -0.1619259 -0.01536794
## [6,] -0.3734771 0.17194306 0.45373418 0.191260029 0.1874822 -0.08377237
## [7,] 0.2243508 0.48404435 0.62812782 -0.030329127 0.1482495 0.25752940
## [8,] 0.2094749 -0.55078264 0.20658376 -0.282381831 0.5624860 -0.32298239
```

```
## [9,] -0.2445807 -0.48431310 0.46412069 -0.214492216 -0.3997820 0.35706914
##      [,7]      [,8]      [,9]
## [1,] 0.38138068 0.12465987 -0.11492862
## [2,] 0.15893251 -0.81032177 -0.16266295
## [3,] 0.18233095 0.06416707 0.66190812
## [4,] -0.69620751 0.16573993 -0.25177306
## [5,] -0.04767957 -0.13505066 -0.03809096
## [6,] 0.42777608 0.19839375 -0.56918844
## [7,] -0.27622581 -0.35613350 0.16873731
## [8,] 0.08555707 -0.31636479 -0.04719694
## [9,] 0.20604210 0.10832772 0.32045892
```

Una vez realizada la función es posible apreciar la matriz de autovalores, al igual que la de autovectores, cabe resaltar que es importante separarlas.

##- Valores propios

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

```
## [1] 5.65593947 2.08210029 0.50421482 0.26502753 0.18315864 0.12379319 0.10506192
## [8] 0.05851375 0.02219038
```

Vectores propios

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

```
##      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.3931477 -0.02753861 -0.22119309 -0.006126378 0.3207620 0.72015586
## [2,] -0.4025537 -0.01570975 -0.25231615 0.040700251 -0.1171397 0.22432550
## [3,] -0.3973528 0.08888469 -0.07825139 0.339493732 0.4867849 -0.01967516
## [4,] -0.3670814 -0.26941371 -0.01721159 0.068300993 0.2947317 0.35394225
## [5,] 0.3118165 -0.34165268 0.14995507 0.845658485 -0.1619259 -0.01536794
## [6,] -0.3734771 0.17194306 0.45373418 0.191260029 0.1874822 -0.08377237
## [7,] 0.2243508 0.48404435 0.62812782 -0.030329127 0.1482495 0.25752940
## [8,] 0.2094749 -0.55078264 0.20658376 -0.282381831 0.5624860 -0.32298239
## [9,] -0.2445807 -0.48431310 0.46412069 -0.214492216 -0.3997820 0.35706914
##      [,7]      [,8]      [,9]
## [1,] 0.38138068 0.12465987 -0.11492862
## [2,] 0.15893251 -0.81032177 -0.16266295
## [3,] 0.18233095 0.06416707 0.66190812
## [4,] -0.69620751 0.16573993 -0.25177306
## [5,] -0.04767957 -0.13505066 -0.03809096
## [6,] 0.42777608 0.19839375 -0.56918844
## [7,] -0.27622581 -0.35613350 0.16873731
## [8,] 0.08555707 -0.31636479 -0.04719694
## [9,] 0.20604210 0.10832772 0.32045892
```

Calcular la proporcion de variabilidad

Se obtiene dividiendo los valores propios entre la suma de los valores propios

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

```
## [1] 0.628437719 0.231344477 0.056023869 0.029447503 0.020350960 0.013754799
## [7] 0.011673547 0.006501528 0.002465598
```

Calcular la proporción de variabilidad acumulada

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

```
## [1] 0.6284377 0.8597822 0.9158061 0.9452536 0.9656045 0.9793593 0.9910329
## [8] 0.9975344 1.0000000
```

En este caso, una vez los resultados fueron observados, escogeremos los 2 primeros factores, ya que, son los que mejor representan la variabilidad acumulada.

0.6284377 0.8597822

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,]  0.9349924 -0.03973679 -0.15706498
## [2,] -0.9573620 -0.02266837 -0.17916487
## [3,] -0.9449932  0.12825603 -0.05556481
## [4,] -0.8730011 -0.38875010 -0.01222162
## [5,]  0.7415688 -0.49298721  0.10648022
## [6,] -0.8882114  0.24810498  0.32218796
## [7,]  0.5335561  0.69845105  0.44602154
## [8,]  0.4981777 -0.79475097  0.14669117
## [9,] -0.5816671 -0.69883885  0.32956322
```

Rotación varimax

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

```
## $loadings
##
## Loadings:
##           [,1]      [,2]      [,3]
## [1,]  0.708  0.564  0.284
## [2,] -0.717 -0.355 -0.555
## [3,] -0.795 -0.372 -0.377
## [4,] -0.397 -0.602 -0.628
```

```
## [5,] 0.888      0.101
## [6,] -0.794 -0.569
## [7,]      0.252 0.953
## [8,] 0.910 -0.219 -0.161
## [9,]      -0.850 -0.458
##
##           [,1] [,2] [,3]
## SS loadings 4.056 2.106 2.080
## Proportion Var 0.451 0.234 0.231
## Cumulative Var 0.451 0.685 0.916
##
## $rotmat
##           [,1] [,2] [,3]
## [1,] 0.7461406 0.4983919 0.4414520
## [2,] -0.6581439 0.4519395 0.6021605
## [3,] 0.1006023 -0.7398353 0.6652240
```

En los “Loadings” se puede observar en la variable 1, una mayor carga en el factor 1, en la variable 2 todos los factores cargan de manera negativa, siendo la más grande el factor 1, igual las variables 3 y 4 cargan de manera negativa.

En el apartado de “SSloadings” podemos notar que la proporción de variabilidad para el factor 1 es de 45 %, en el factor 2 un 23% y en el factor 3 tambien un 23%, por lo que, podemos concluir que el factor 1 y 2 son los cuales explican la mayor cantidad de variabilidad.

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,] 0.0995408 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000 0.00000000
## [2,] 0.00000000 0.05084412 0.00000000 0.0000000 0.00000000 0.00000000 0.00000000
## [3,] 0.00000000 0.00000000 0.08745088 0.0000000 0.00000000 0.00000000 0.00000000
## [4,] 0.00000000 0.00000000 0.00000000 0.086593 0.00000000 0.00000000 0.00000000
## [5,] 0.00000000 0.00000000 0.00000000 0.0000000 0.1957012 0.00000000 0.00000000
## [6,] 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000 0.0457193 0.00000000
## [7,] 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000 0.02854884
## [8,] 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000 0.00000000
## [9,] 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000 0.00000000
##           [,8] [,9]
## [1,] 0.00000000 0.00000000
## [2,] 0.00000000 0.00000000
## [3,] 0.00000000 0.00000000
## [4,] 0.00000000 0.00000000
## [5,] 0.00000000 0.00000000
## [6,] 0.00000000 0.00000000
## [7,] 0.00000000 0.00000000
## [8,] 0.09867157 0.00000000
## [9,] 0.00000000 0.06467568
```

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

```
RP<-R-Psi.est.1  
head(RP)
```

```
##          millas  cilindros cilindrada  caballos      eje      peso  
## millas      0.9004592 -0.8521620 -0.8475514 -0.7761684  0.6811719 -0.8676594  
## cilindros -0.8521620  0.9491559  0.9020329  0.8324475 -0.6999381  0.7824958  
## cilindrada -0.8475514  0.9020329  0.9125491  0.7909486 -0.7102139  0.8879799  
## caballos   -0.7761684  0.8324475  0.7909486  0.9134070 -0.4487591  0.6587479  
## eje        0.6811719 -0.6999381 -0.7102139 -0.4487591  0.8042988 -0.7124406  
## peso      -0.8676594  0.7824958  0.8879799  0.6587479 -0.7124406  0.9542807  
##          velocidad  cambios carburadores  
## millas      0.41868403  0.4802848   -0.5509251  
## cilindros -0.59124207 -0.4926866    0.5269883  
## cilindrada -0.43369788 -0.5555692    0.3949769  
## caballos   -0.70822339 -0.1257043    0.7498125  
## eje        0.09120476  0.6996101   -0.0907898  
## peso      -0.17471588 -0.5832870    0.4276059
```

Calculo de la matriz de autovalores y autovectores

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

Autovalores

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

```
## [1]  5.572165151  2.000038362  0.453056211  0.119348645  0.080093612  
## [6]  0.045648492  0.026571256 -0.005172587 -0.049494554
```

Autovectores

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

```
## [1]  5.572165151  2.000038362  0.453056211  0.119348645  0.080093612  
## [6]  0.045648492  0.026571256 -0.005172587 -0.049494554
```

Proporcion de variabilidad

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

```
## [1]  0.6760486577  0.2426567076  0.0549675100  0.0144800969  0.0097174397  
## [6]  0.0055383502  0.0032237850 -0.0006275695 -0.0060049775
```

Proporcion de variabilidad acumulada

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



```
## [1] 0.6760487 0.9187054 0.9736729 0.9881530 0.9978704 1.0034088 1.0066325
## [8] 1.0060050 1.0000000
```

Al observar los resultados podemos escoger los 2 primeros factores, **0.6760487**, **0.9187054**.

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,]  0.9252825 -0.04770735 -0.1292947970
## [2,] -0.9559189 -0.01514217 -0.1700268077
## [3,] -0.9370706  0.13428909 -0.0550328186
## [4,] -0.8672733 -0.37209492  0.0006824083
## [5,]  0.7196227 -0.46095442  0.0796155549
## [6,] -0.8869184  0.25806723  0.2987270323
## [7,]  0.5371042  0.70107682  0.4249817837
## [8,]  0.4907391 -0.77625561  0.1512985675
## [9,] -0.5815371 -0.68402174  0.3245360152
```

Rotacion varimax

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

```
## $loadings
##
## Loadings:
##           [,1]      [,2]      [,3]
## [1,]  0.738  0.497  0.290
## [2,] -0.740 -0.319 -0.541
## [3,] -0.813 -0.323 -0.366
## [4,] -0.436 -0.580 -0.604
## [5,]  0.852
## [6,] -0.830 -0.503
## [7,]           0.273  0.941
## [8,]  0.879 -0.266 -0.153
## [9,]          -0.842 -0.451
##
##           [,1]      [,2]      [,3]
## SS loadings  4.130  1.898  1.997
## Proportion Var 0.459 0.211 0.222
## Cumulative Var 0.459 0.670 0.892
##
## $rotmat
##           [,1]      [,2]      [,3]
## [1,]  0.77336100  0.4585853  0.4377354
## [2,] -0.63083151  0.4880770  0.6031852
## [3,]  0.06296326 -0.7426172  0.6667498
```

En esta matriz en nuestros “loadings” podemos observar que para la variable 1, carga mayormente en el factor 1 y menos en el factor 3, en la variable 2, todos los factores carga de manera negativa, para la variable 3 igual que la variable 2, todos los factores cargan de manera negativa.

Proseguimos con el apartado “SSloadings” obtenemos que el factor 1 explica un 45%, el factor 2 explica un 21% y el factor 3 un 22%, por lo que podemos concluir que los factores 1 y 3 son los cuales 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,] 0.1248591 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.05708058 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.00000000 0.1008365 0.0000000 0.0000000 0.0000000 0.0000000
## [4,] 0.0000000 0.00000000 0.0000000 0.109382 0.0000000 0.0000000 0.0000000
## [5,] 0.0000000 0.00000000 0.0000000 0.0000000 0.2633256 0.0000000 0.0000000
## [6,] 0.0000000 0.00000000 0.0000000 0.0000000 0.0000000 0.0575393 0.0000000
## [7,] 0.0000000 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.03940087
## [8,] 0.0000000 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
## [9,] 0.0000000 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##           [,8]      [,9]
## [1,] 0.0000000 0.00000000
## [2,] 0.0000000 0.00000000
## [3,] 0.0000000 0.00000000
## [4,] 0.0000000 0.00000000
## [5,] 0.0000000 0.00000000
## [6,] 0.0000000 0.00000000
## [7,] 0.0000000 0.00000000
## [8,] 0.1337111 0.00000000
## [9,] 0.0000000 0.08860526
```

Obtención 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]
## Mazda RX4      2.278446  0.12939676 -0.4187558
## Mazda RX4 Wag  2.066387  0.05997537 -0.1275513
## Datsun 710      3.755690  3.04922364  2.5989277
## Hornet 4 Drive -1.518807  1.71740014  1.7880288
## Hornet Sportabout -3.676876 -0.79790863 -1.4339938
## Valiant        -2.403576  1.47802676  2.1266113
```

PFA

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

```
##           [,1]      [,2]      [,3]
## Mazda RX4      2.250309 -0.007388463 -0.4237748
## Mazda RX4 Wag  2.033987 -0.052856971 -0.1374758
```

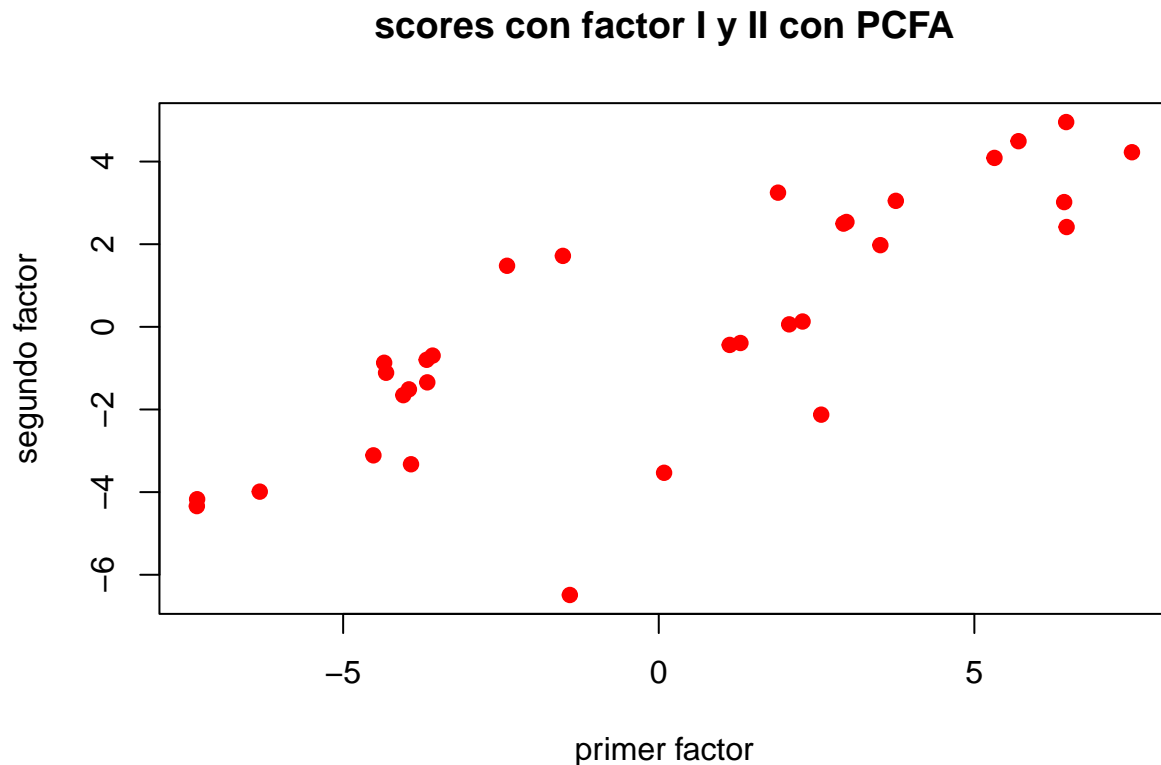
```
## Datsun 710      3.919409  2.819466334  2.5434606
## Hornet 4 Drive -1.350609  1.772825496  1.7615711
## Hornet Sportabout -3.670183 -0.623580325 -1.3973205
## Valiant        -2.224175  1.594832408  2.0928410
```

graficamos ambos scores

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

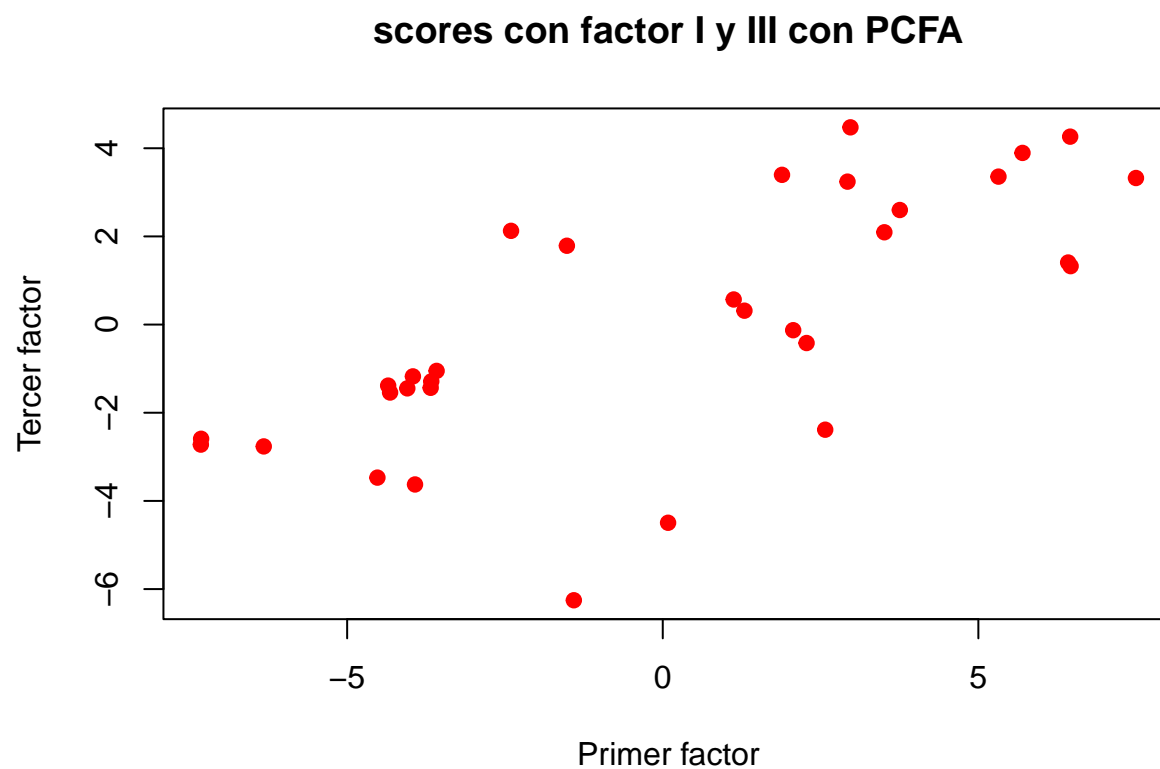
Factor I y II

```
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="red")
```



Factor I y III

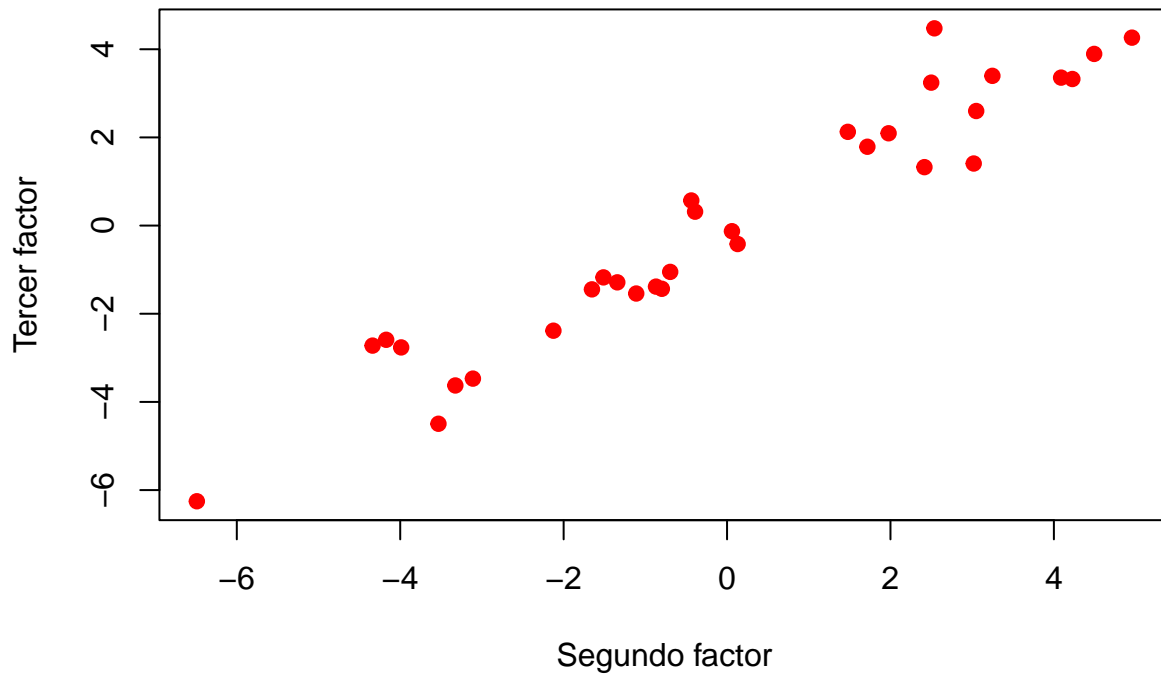
```
pl2<-plot(FS.est.1[,1], FS.est.1[,3], xlab="Primer factor",
          ylab="Tercer factor", main="scores con factor I y III con PCFA",
          pch=19, col="red")
```



Factor II y III

```
p13<-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="red")
```

scores con factor II y III con PCFA



Es posible apreciar que, entre los graficos de los factores se observar que se presenta una mayor separación en nuestros datos en el gráfico de los factores (**I y III**).

Estos factores, de igual manera y como ya se ha mencionado fueron los cuales representaban mejor nuestra variabilidad.

Gráfico con nombre de nustras variables

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

Factor I y III

```
p12<-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="red")
text(FS.est.1[,1], FS.est.1[,3], labels = rownames(x), pos=4, col="black")
```

scores con factor I y III con PCFA

