Evaluación MLI: Ejercicio 2

(Reducción de la dimensionalidad)

Inmaculada Perea Fernández

Abril 2017

Acceder a los datos gironde la librería *PCAmixdata*. En los siguientes apartados seleccionar los registros completos si hay valores perdidos.

Carga e instalación de librerías necesarias

```
if (!require('cluster')) install.packages('cluster'); library('cluster')
if (!require('PCAmixdata')) install.packages('PCAmixdata'); library('PCAmixdata')
if (!require('corrplot')) install.packages('corrplot'); library('corrplot')

# Necesarias para la normalización
if (!require('Rcpp')) install.packages('Rcpp'); library('Rcpp')
if (!require('clusterSim')) install.packages('clusterSim'); library('clusterSim')
if (!require('digest')) install.packages('digest'); library('digest')


if (!require('GA')) install.packages('GA'); library('GA')
if (!require('leaps')) install.packages('leaps'); library('leaps')
```

Ejercicio 2.1

Realizar e interpretar un análisis de componentes principales (matriz de correlaciones) para *gironde$employment*.

2.1.1 Carga, inspección y preparación de los datos

Carga de los datos

```
data(gironde)
employment.na<-gironde$employment
head(employment.na)
```

```
##                    farmers tradesmen managers workers unemployed
## ABZAC                 1.98      3.68     3.97   38.25      13.60
## AILLAS                5.23      5.23     1.96   21.57      15.03
## AMBARES-ET-LAGRAVE    0.10      4.38     5.56   35.98      18.23
## AMBES                 0.18      2.29     3.70   42.42      15.11
## ANDERNOS-LES-BAINS    0.30      3.80     8.19   18.65      13.04
## ANGLADE               3.13      5.63     1.25   39.37      16.87
##                    middleempl retired employrate   income
## ABZAC                    9.63   28.90      89.26 17670.60
## AILLAS                  14.38   36.60      90.88 19422.49
## AMBARES-ET-LAGRAVE      15.48   20.28      90.25 21047.07
## AMBES                    8.98   27.33      87.38 18014.52
## ANDERNOS-LES-BAINS      12.07   43.97      89.43 27147.48
## ANGLADE                  5.63   28.12      88.71 15897.99
```

```r
str(employment.na)
```

```
## 'data.frame':    542 obs. of  9 variables:
##  $ farmers   : num  1.98 5.23 0.1 0.18 0.3 ...
##  $ tradesmen : num  3.68 5.23 4.38 2.29 3.8 5.63 4.21 1.75 4.61 2.3 ...
##  $ managers  : num  3.97 1.96 5.56 3.7 8.19 1.25 4.21 3.51 5.8 0 ...
##  $ workers   : num  38.2 21.6 36 42.4 18.6 ...
##  $ unemployed: num  13.6 15 18.2 15.1 13 ...
##  $ middleempl: num  9.63 14.38 15.48 8.98 12.07 ...
##  $ retired   : num  28.9 36.6 20.3 27.3 44 ...
##  $ employrate: num  89.3 90.9 90.2 87.4 89.4 ...
##  $ income    : num  17671 19422 21047 18015 27147 ...
```

```r
summary(employment.na)
```

```
##     farmers          tradesmen         managers         workers
##  Min.   : 0.0000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00
##  1st Qu.: 0.5125   1st Qu.: 2.772   1st Qu.: 2.795   1st Qu.:28.57
##  Median : 1.9700   Median : 3.995   Median : 4.650   Median :33.66
##  Mean   : 3.4650   Mean   : 4.189   Mean   : 5.287   Mean   :33.52
##  3rd Qu.: 4.6875   3rd Qu.: 5.300   3rd Qu.: 7.147   3rd Qu.:38.40
##  Max.   :33.3300   Max.   :16.130   Max.   :22.730   Max.   :57.14
##
##    unemployed      middleempl        retired        employrate
##  Min.   : 0.00   Min.   : 0.000   Min.   : 9.33   Min.   : 75.08
##  1st Qu.:11.22   1st Qu.: 8.523   1st Qu.:23.25   1st Qu.: 88.35
##  Median :13.55   Median :11.875   Median :27.45   Median : 90.66
##  Mean   :13.38   Mean   :11.993   Mean   :28.17   Mean   : 90.30
##  3rd Qu.:15.59   3rd Qu.:15.440   3rd Qu.:32.14   3rd Qu.: 92.71
##  Max.   :33.33   Max.   :31.580   Max.   :51.28   Max.   :100.00
##
##      income
##  Min.   :12187
##  1st Qu.:18367
##  Median :19990
##  Mean   :21003
##  3rd Qu.:22768
##  Max.   :70062
##  NA's   :2
```

```r
dim(employment.na)
```

```
## [1] 542   9
```

Eliminación de los valores perdidos

```r
employment<-na.omit(employment.na)
dim(employment)
```

```
## [1] 540   9
```

```r
summary(employment)
```

```
##     farmers          tradesmen         managers         workers
##  Min.   : 0.0000   Min.   : 0.000   Min.   : 0.000   Min.   : 7.69
##  1st Qu.: 0.5025   1st Qu.: 2.780   1st Qu.: 2.825   1st Qu.:28.64
##  Median : 1.9550   Median : 4.000   Median : 4.650   Median :33.67
```

```
##  Mean   : 3.3544   Mean   : 4.204   Mean   : 5.286   Mean   :33.65
##  3rd Qu.: 4.6125   3rd Qu.: 5.312   3rd Qu.: 7.143   3rd Qu.:38.41
##  Max.   :29.0300   Max.   :16.130   Max.   :22.730   Max.   :57.14
##    unemployed      middleempl        retired        employrate
##  Min.   : 0.00   Min.   : 0.000   Min.   : 9.33   Min.   : 75.08
##  1st Qu.:11.23   1st Qu.: 8.547   1st Qu.:23.23   1st Qu.: 88.35
##  Median :13.55   Median :11.905   Median :27.45   Median : 90.66
##  Mean   :13.35   Mean   :12.005   Mean   :28.16   Mean   : 90.31
##  3rd Qu.:15.55   3rd Qu.:15.465   3rd Qu.:32.14   3rd Qu.: 92.70
##  Max.   :29.19   Max.   :31.580   Max.   :51.28   Max.   :100.00
##      income
##  Min.   :12187
##  1st Qu.:18367
##  Median :19990
##  Mean   :21003
##  3rd Qu.:22768
##  Max.   :70062
```

Estandarización de los datos

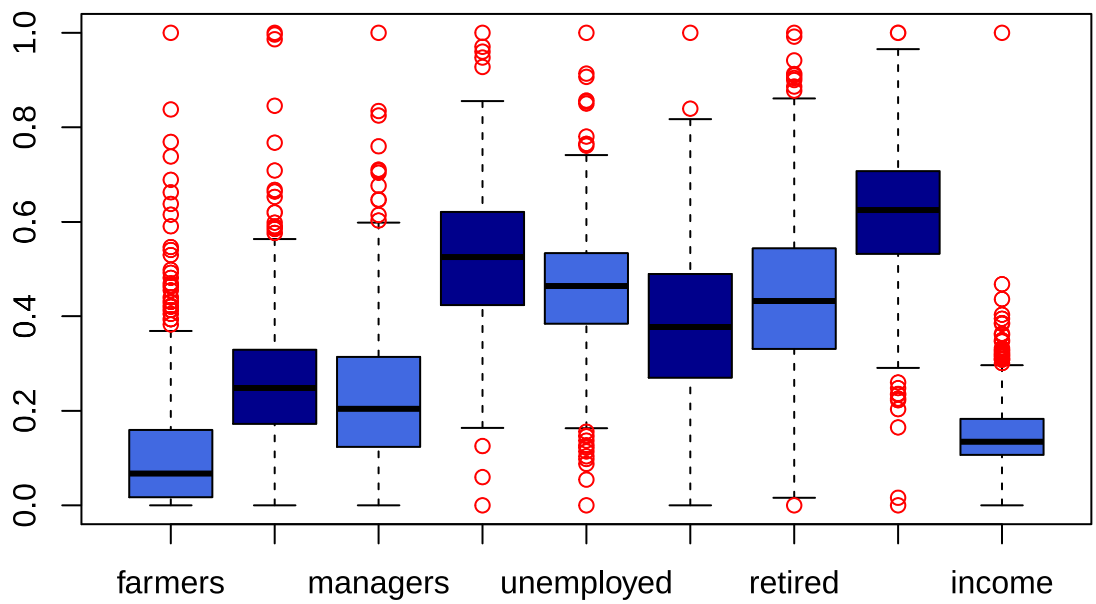
Existe mucha variabilidad con income y el resto de variables, al tratarse de atributos cuantitativos es recomendable tipificar para que no existan problemas de escala.

```
# Normalización a través del criterio min-max
norm.employment=data.Normalization (employment, type="n4", normalization="column")
summary(norm.employment)
```

```
##     farmers          tradesmen         managers          workers
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.01731   1st Qu.:0.1723   1st Qu.:0.1243   1st Qu.:0.4237
##  Median :0.06734   Median :0.2480   Median :0.2046   Median :0.5254
##  Mean   :0.11555   Mean   :0.2606   Mean   :0.2325   Mean   :0.5249
##  3rd Qu.:0.15889   3rd Qu.:0.3294   3rd Qu.:0.3142   3rd Qu.:0.6211
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##    unemployed       middleempl         retired         employrate
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.3848   1st Qu.:0.2707   1st Qu.:0.3313   1st Qu.:0.5325
##  Median :0.4642   Median :0.3770   Median :0.4319   Median :0.6252
##  Mean   :0.4572   Mean   :0.3801   Mean   :0.4488   Mean   :0.6111
##  3rd Qu.:0.5329   3rd Qu.:0.4897   3rd Qu.:0.5437   3rd Qu.:0.7070
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##      income
##  Min.   :0.0000
##  1st Qu.:0.1068
##  Median :0.1348
##  Mean   :0.1523
##  3rd Qu.:0.1828
##  Max.   :1.0000
```

Diagrama de caja

```
boxplot(norm.employment, col=c("royalblue", "darkblue"), outcol="red")
```



Cálculo de la matriz de correlaciones

```
R<- cor(norm.employment)
round(R,2)
```

```
##            farmers tradesmen managers workers unemployed middleempl
## farmers       1.00     -0.06    -0.18   -0.16      -0.23      -0.33
## tradesmen    -0.06      1.00    -0.06   -0.14      -0.10      -0.10
## managers     -0.18     -0.06     1.00   -0.36       0.09       0.31
## workers      -0.16     -0.14    -0.36    1.00      -0.19      -0.19
## unemployed   -0.23     -0.10     0.09   -0.19       1.00      -0.03
## middleempl   -0.33     -0.10     0.31   -0.19      -0.03       1.00
## retired       0.04      0.00    -0.26   -0.48      -0.22      -0.40
## employrate    0.19      0.03     0.33   -0.07      -0.18       0.24
## income       -0.06      0.05     0.48   -0.24      -0.01       0.38
##            retired employrate income
## farmers       0.04       0.19  -0.06
## tradesmen     0.00       0.03   0.05
## managers     -0.26       0.33   0.48
## workers      -0.48      -0.07  -0.24
## unemployed   -0.22      -0.18  -0.01
## middleempl   -0.40       0.24   0.38
## retired       1.00      -0.27  -0.23
## employrate   -0.27       1.00   0.41
## income       -0.23       0.41   1.00
```

Determinante de la matriz de correlaciones

```r
det(R)
```

```
## [1] 2.93323e-07
```

Observamos que la correlación entre cada 2 variables no es muy elevada, pero que el determinante de la matriz de correlaciones es próximo a 0, lo que indica que las variables están altamente correladas

Representación gráfica de la matriz de correlaciones

```r
corrplot(R, method="ellipse")
```


```r
corrplot(R, method="number")
```

2.1.2 Análisis de componentes principales usando *princomp*

```
employment.acp<- princomp(employment, cor = TRUE) # cor=TRUE para tipificar los datos
summary(employment.acp)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4    Comp.5
## Standard deviation     1.5488296 1.2567129 1.1840160 1.0228749 0.9676437
## Proportion of Variance 0.2665414 0.1754808 0.1557660 0.1162526 0.1040371
## Cumulative Proportion  0.2665414 0.4420222 0.5977882 0.7140408 0.8180779
##                            Comp.6     Comp.7     Comp.8       Comp.9
## Standard deviation     0.79952880 0.71169174 0.70110395 5.951540e-04
## Proportion of Variance 0.07102737 0.05627835 0.05461631 3.935647e-08
## Cumulative Proportion  0.88910531 0.94538366 0.99999996 1.000000e+00
```

Tabla resumen con los valores de interés

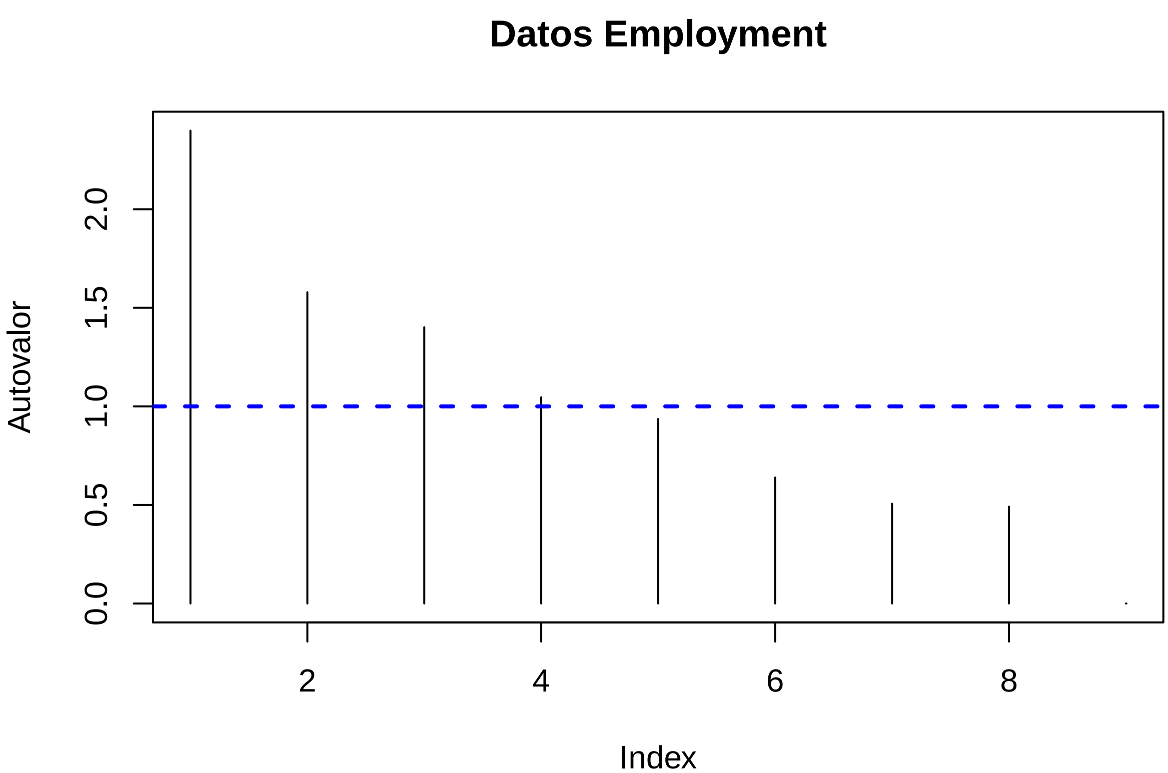
```
resumen<- matrix(NA, nrow=length(employment.acp$sdev), ncol=3)
resumen[,1]<-  employment.acp$sdev^2
resumen[,2]<- 100*resumen[,1]/sum(resumen[,1])
resumen[,3]<- cumsum(resumen[,2])
colnames(resumen)<- c("Autovalor","Porcentaje","Porcentaje acumulado")
round(resumen, 4)
```

```
##       Autovalor Porcentaje Porcentaje acumulado
##  [1,]    2.3989    26.6541              26.6541
```

```
## [2,]    1.5793    17.5481             44.2022
## [3,]    1.4019    15.5766             59.7788
## [4,]    1.0463    11.6253             71.4041
## [5,]    0.9363    10.4037             81.8078
## [6,]    0.6392     7.1027             88.9105
## [7,]    0.5065     5.6278             94.5384
## [8,]    0.4915     5.4616            100.0000
## [9,]    0.0000     0.0000            100.0000
```

Gráfico de sedimentación

```r
plot(resumen[,1], type="h", main="Datos Employment", ylab="Autovalor")
abline(h=mean(resumen[,1]), lwd=2, lty=2, col="blue")
```



2.1.3 Selección del número de componentes principales

Existen diferentes criterios para seleccionar el número de componentes principales:

1) Porcentaje acumulado mayor que un umbral

Si tomamos como umbral el *80%*, entonces tomaríamos las 5 primeras componentes principales.

2) Autovalores superiores a la media

Si seguimos este criterio también nos quedaríamos con las 4 primeras componenentes principales, que son las que presentan autovalores mayor a la media (1)

3) Mediante contrastes de hipótesis

En primer lugar comprobamos normalidad multivariante como condición para utilizar este método inferencial

```
source("Test_Mardia.r")
Test_Mardia(employment)
```

```
## $g1p
## [1] 39.55754
##
## $chi.skew
## [1] 3560.178
##
## $p.value.skew
## [1] 0
##
## $chi.small.skew
## [1] 3583.932
##
## $p.value.small
## [1] 0
##
## $g2p
## [1] 190.9438
##
## $z.kurtosis
## [1] 75.92012
##
## $p.value.kurt
## [1] 0
```

Obtenemos: *p.value.skew*, *p.value.small* y *p.value.kurt* igual a 0. Por tanto, no se acepta la normalidad multivariante, esto implica que no es posible seleccionar el número de componentes principales usando método inferencial.

Coeficientes que definen la combinación lineal de las variables y las componentes principales

```
round(loadings(employment.acp), 3)
```

```
##
## Loadings:
##            Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## farmers     0.137  0.341  0.502 -0.410  0.348  0.334  0.203 -0.276 -0.309
## tradesmen          0.189         0.835  0.466                      -0.177
## managers   -0.489  0.140 -0.140 -0.114        -0.511 -0.182 -0.587 -0.262
## workers     0.170 -0.657  0.354               -0.295  0.108        -0.549
## unemployed        -0.180 -0.574 -0.309  0.588                0.302 -0.297
## middleempl -0.457 -0.144         0.106 -0.398  0.671               -0.364
## retired     0.332  0.549 -0.209        -0.378 -0.178         0.291 -0.531
## employrate -0.382  0.135  0.465         0.112        -0.576  0.514
## income     -0.492  0.157                      -0.184  0.756  0.347
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## SS loadings     0.998  1.001  1.001  0.999  1.001  1.000  1.001  1.000
## Proportion Var  0.111  0.111  0.111  0.111  0.111  0.111  0.111  0.111
## Cumulative Var  0.111  0.222  0.333  0.444  0.556  0.667  0.778  0.889
##                Comp.9
## SS loadings     1.000
```

```
## Proportion Var  0.111
## Cumulative Var  1.000
```

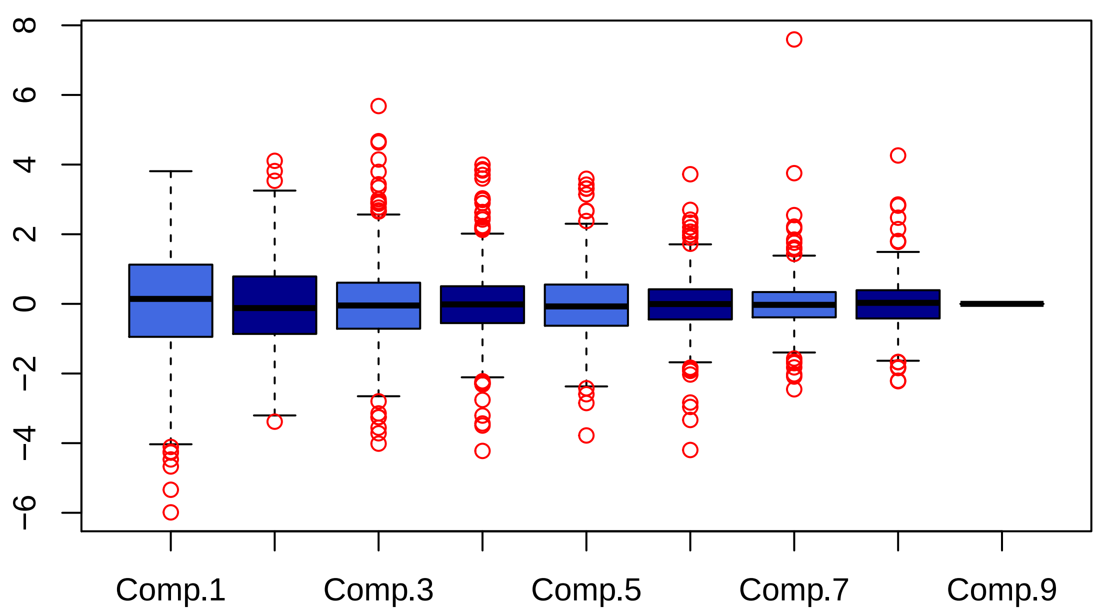
correlaciones entre las variables y la componentes

```
correlaciones<-loadings(employment.acp)%*%diag(employment.acp$sdev)
round(correlaciones, 3)
```

```
##               [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8] [,9]
## farmers      0.213  0.428  0.594 -0.420  0.336  0.267  0.144 -0.193    0
## tradesmen    0.015  0.237  0.028  0.854  0.451  0.069 -0.026 -0.069    0
## managers    -0.758  0.175 -0.166 -0.116  0.022 -0.409 -0.130 -0.412    0
## workers      0.264 -0.826  0.419  0.090 -0.043 -0.236  0.077  0.047    0
## unemployed -0.103 -0.227 -0.680 -0.316  0.569  0.075 -0.043  0.212    0
## middleempl -0.709 -0.181 -0.115  0.108 -0.386  0.536 -0.045 -0.047    0
## retired      0.515  0.690 -0.247  0.028 -0.366 -0.142 -0.035  0.204    0
## employrate -0.592  0.170  0.550 -0.058  0.109 -0.073 -0.410  0.360    0
## income      -0.762  0.197  0.081  0.054  0.015 -0.147  0.538  0.244    0
```

Representación gráfica de la variabilidad de las puntuaciones de las componentes principales

```
boxplot(employment.acp$scores,
        col=c("royalblue", "darkblue"),
        outcol="red", notched=TRUE)
```



Observamos que la varianza va decreciendo

2.1.3.1 Representación con 4 componentes principales

Cálculo de los autovalores y autovectores

```
descompespec<-eigen(R)
autovalores<- descompespec$values
autovectores<- descompespec$vectors
```

Comunalidades con 4 componentes principales

Comunalidades para cada variable, es la suma de correlaciones cuadrado con las c.p. seleccionadas

```
cbind(apply(correlaciones[,1:4]^2, 1, sum))
```

```
##                 [,1]
## farmers    0.7574830
## tradesmen  0.7868471
## managers   0.6460763
## workers    0.9344505
## unemployed 0.6242943
## middleempl 0.5595442
## retired    0.8029337
## employrate 0.6850632
## income     0.6296747
```

Las comunalidades para 4 componentes no son bajas, por lo que todas las variables quedan explicadas con 4 CP. El caso más desfavorable es del de la variable *middleempl*, con una comunalidad *0.56*

Correlaciones reproducidas con 4 componentes principales

```
#Matriz de correlaciones reproducidas
Raprox4<- autovectores[,1:4]%*%diag(autovalores[1:4])%*%t(autovectores[,1:4])
```

Correlación residual con 4 componentes

```
Resid4 = R - Raprox4
corrplot(Resid4)
```


```
mean((Resid4)^2)
```

```
## [1] 0.02201881
```

2.1.3.2 Representación con 5 componentes principales

Comunalidades con 5 componentes principales

```
cbind(apply(correlaciones[,1:5]^2, 1, sum))
```

```
##                 [,1]
## farmers    0.8705586
## tradesmen  0.9898055
## managers   0.6465656
## workers    0.9363263
## unemployed 0.9475038
## middleempl 0.7081657
## retired    0.9369789
## employrate 0.6968875
## income     0.6299096
```

Correlaciones reproducidas con 5 componentes principales

```
#Matriz de correlaciones reproducidas
Raprox5<- autovectores[,1:5]%*%diag(autovalores[1:5])%*%t(autovectores[,1:5])
```

Correlación residual con 5 componentes

```r
Resid5 = R- Raprox5
mean( (Resid5) ^ 2 )
```

```
## [1] 0.01119508
```

```r
corrplot(Resid5)
```

Observamos que con 5 CP las variable originales quedan mejor explicadas, pero nos podemos quedar con 4 CP porque tambien se obtienen resultados aceptables. Las correlaciones residuales con 4 y 5 componentes tambien disminuye de 0.0220 a 0.0112 respectivamente.

2.1.4 Rotación ortogonal varimax

```r
acprot<- varimax(loadings(employment.acp)[,1:4])
summary(acprot)
```

```
##          Length Class    Mode
## loadings 36     loadings numeric
## rotmat   16     -none-   numeric
```

```r
loadings(acprot)
```

```
##
## Loadings:
##            Comp.1 Comp.2 Comp.3 Comp.4
## farmers     0.140  0.149  0.677 -0.233
## tradesmen                -0.114  0.843
```

```
## managers   -0.511  0.115        -0.127
## workers     0.286 -0.713
## unemployed               -0.499 -0.445
## middleempl -0.442 -0.169 -0.157
## retired     0.234  0.627
## employrate -0.355 -0.130  0.488
## income     -0.503         0.122
##
##                 Comp.1 Comp.2 Comp.3 Comp.4
## SS loadings      1.000  1.000  1.000  1.000
## Proportion Var  0.111  0.111  0.111  0.111
## Cumulative Var  0.111  0.222  0.333  0.444
```

Puntuaciones de las componentes rotadas

```
punturota<- employment.acp$scores[,1:4]%*%acprot$rotmat
```

Correlaciones entre las variables y las 4 componentes seleccionadas rotadas

```
corr_rot=cor(employment, punturota)
round(corr_rot, 4)
```

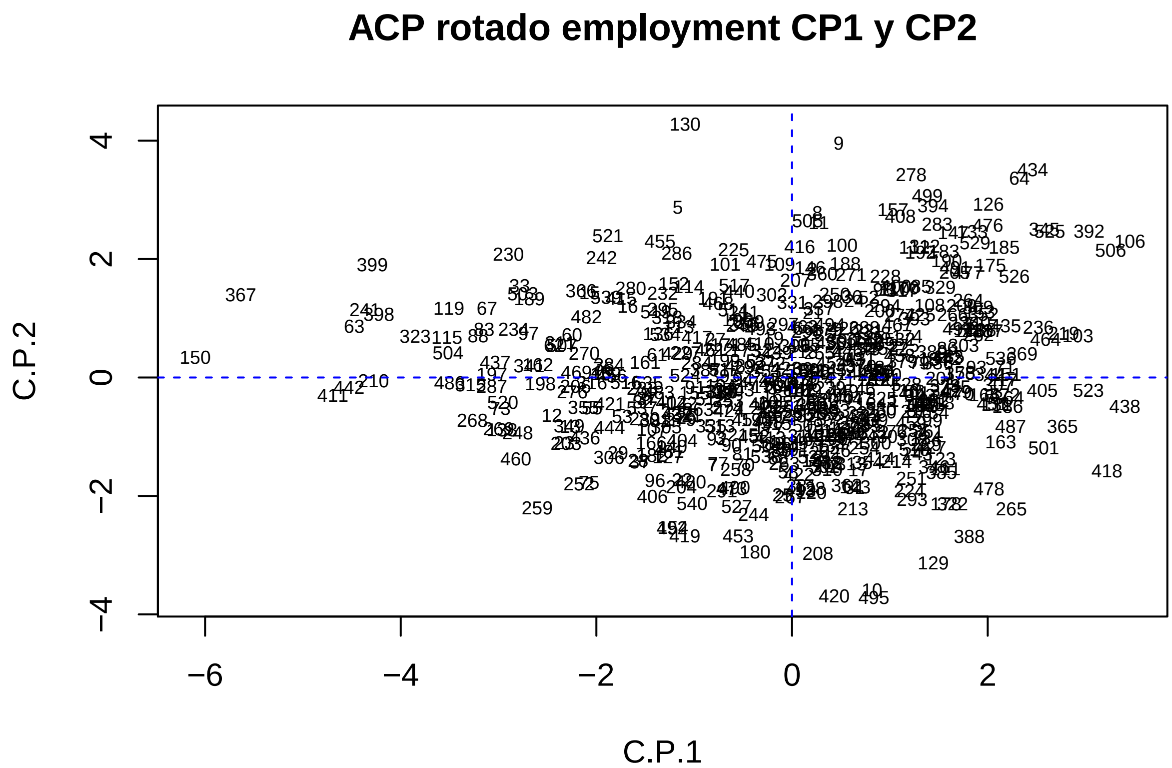
```
##               [,1]    [,2]    [,3]    [,4]
## farmers     0.2071  0.2180  0.7867 -0.1651
## tradesmen  -0.0223  0.1466 -0.0613  0.8689
## managers   -0.7800  0.0818 -0.0225 -0.1440
## workers     0.3802 -0.8585 -0.0054  0.0156
## unemployed -0.1161  0.0669 -0.6220 -0.5057
## middleempl -0.6911 -0.2624 -0.1768  0.0072
## retired     0.4166  0.8141  0.0483  0.1405
## employrate -0.5696 -0.1780  0.5880  0.0777
## income     -0.7737 -0.0086  0.1675  0.0773
```

```
round(correlaciones[,1:4], 4)
```

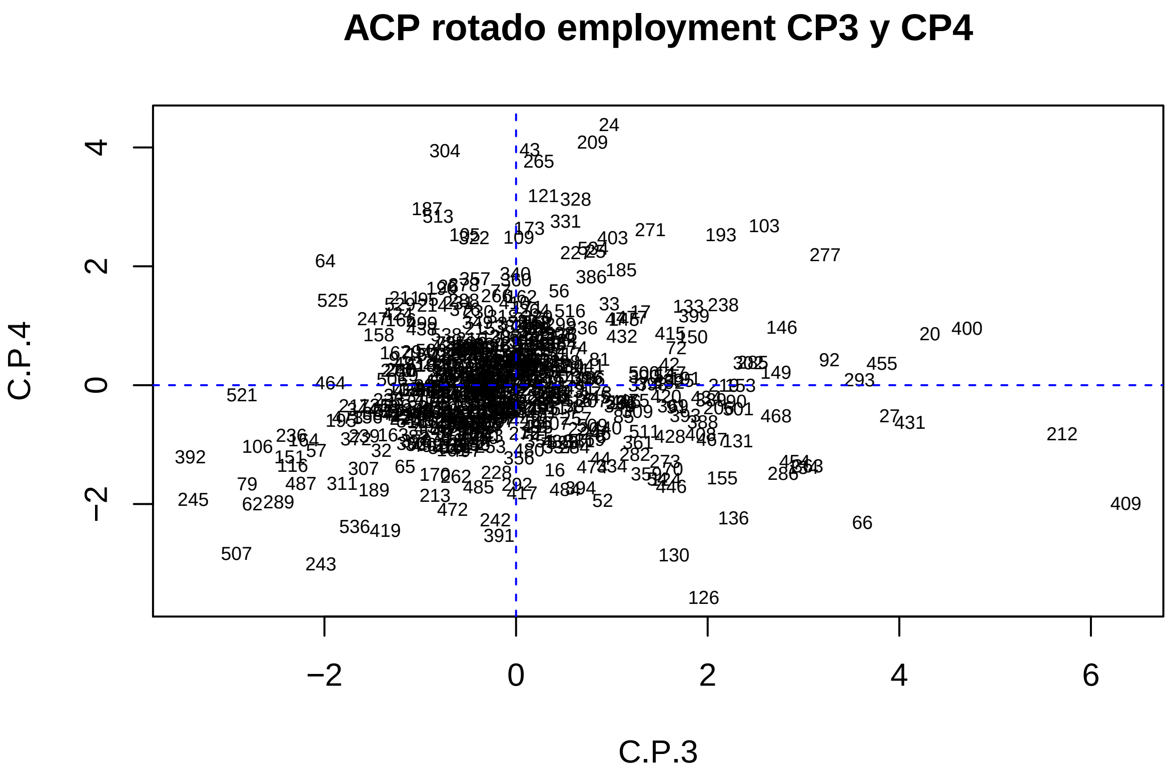
```
##               [,1]    [,2]    [,3]    [,4]
## farmers     0.2126  0.4282  0.5940 -0.4196
## tradesmen   0.0153  0.2369  0.0282  0.8542
## managers   -0.7578  0.1754 -0.1660 -0.1164
## workers     0.2639 -0.8255  0.4186  0.0902
## unemployed -0.1034 -0.2268 -0.6797 -0.3164
## middleempl -0.7086 -0.1807 -0.1147  0.1080
## retired     0.5150  0.6898 -0.2472  0.0279
## employrate -0.5918  0.1695  0.5501 -0.0584
## income     -0.7625  0.1972  0.0809  0.0536
```

Representación gráfica

```
plot(punturota[,1],punturota[,2], type="n",
     main ="ACP rotado employment CP1 y CP2",
     xlab="C.P.1", ylab="C.P.2")

text(punturota[,1], punturota[,2], cex=0.6)

abline(h=0, v=0, lty=2, col="blue")
```




```r
plot(punturota[,3],punturota[,4], type="n",
     main ="ACP rotado employment CP3 y CP4",
     xlab="C.P.3", ylab="C.P.4")

text(punturota[,3], punturota[,4], cex=0.6)

abline(h=0, v=0, lty=2, col="blue")
```



Ejercicio 2.2

Realizar e interpretar un análisis de componentes principales para datos mixtos sobre la unión de $girondeemployment * y * girondeservices$

2.2.1 Carga, inspección y preparación de los datos

Carga de los datos

```
data(gironde)
services<-gironde$services
head(services)
```

```
##                     butcher  baker postoffice dentist grocery nursery
## ABZAC                     0 2 or +     1 or +       0       0       0
## AILLAS                    0      0          0       0  1 or +       0
## AMBARES-ET-LAGRAVE        1 2 or +     1 or +  3 or +  1 or +  1 or +
## AMBES                     0      1     1 or +  1 to 2  1 or +       0
## ANDERNOS-LES-BAINS   2 or + 2 or +     1 or +  3 or +  1 or +       0
## ANGLADE                   0      1          0       0  1 or +       0
##                     doctor chemist restaurant
## ABZAC                    0       1          1
## AILLAS              3 or +       0          1
```

```
## AMBARES-ET-LAGRAVE 3 or +  2 or +     3 or +
## AMBES              3 or +       1     3 or +
## ANDERNOS-LES-BAINS 3 or +  2 or +     3 or +
## ANGLADE                 0       0          2
```

```r
str(services)
```

```
## 'data.frame':    542 obs. of  9 variables:
##  $ butcher   : Factor w/ 3 levels "0","1","2 or +": 1 1 2 1 3 1 1 1 3 1 ...
##  $ baker     : Factor w/ 3 levels "0","1","2 or +": 3 1 3 2 3 2 1 1 3 2 ...
##  $ postoffice: Factor w/ 2 levels "0","1 or +": 2 1 2 2 2 1 1 1 2 1 ...
##  $ dentist   : Factor w/ 3 levels "0","1 to 2","3 or +": 1 1 3 2 3 1 1 1 3 1 ...
##  $ grocery   : Factor w/ 2 levels "0","1 or +": 1 2 2 2 2 2 2 2 2 1 ...
##  $ nursery   : Factor w/ 2 levels "0","1 or +": 1 1 2 1 1 1 1 1 2 1 ...
##  $ doctor    : Factor w/ 3 levels "0","1 to 2","3 or +": 1 3 3 3 3 1 1 1 3 1 ...
##  $ chemist   : Factor w/ 3 levels "0","1","2 or +": 2 1 3 2 3 1 1 1 3 1 ...
##  $ restaurant: Factor w/ 4 levels "0","1","2","3 or +": 2 2 4 4 4 3 3 1 4 3 ...
```

```r
summary(services)
```

```
##    butcher       baker      postoffice    dentist      grocery
##  0     :371   0     :291   0     :346   0     :380   0     :365
##  1     : 95   1     :128   1 or +:196   1 to 2: 90   1 or +:177
##  2 or +: 76   2 or +:123                3 or +: 72
##
##    nursery       doctor       chemist      restaurant
##  0     :520   0     :326   0     :357   0     :247
##  1 or +: 22   1 to 2: 92   1     :107   1     :122
##               3 or +:124   2 or +: 78   2     : 52
##                                         3 or +:121
```

```r
dim(services)
```

```
## [1] 542   9
```

Union de los datos employment y services

```r
mix_data.na=cbind(employment.na, services)
str(mix_data.na)
```

```
## 'data.frame':    542 obs. of  18 variables:
##  $ farmers   : num  1.98 5.23 0.1 0.18 0.3 ...
##  $ tradesmen : num  3.68 5.23 4.38 2.29 3.8 5.63 4.21 1.75 4.61 2.3 ...
##  $ managers  : num  3.97 1.96 5.56 3.7 8.19 1.25 4.21 3.51 5.8 0 ...
##  $ workers   : num  38.2 21.6 36 42.4 18.6 ...
##  $ unemployed: num  13.6 15 18.2 15.1 13 ...
##  $ middleempl: num  9.63 14.38 15.48 8.98 12.07 ...
##  $ retired   : num  28.9 36.6 20.3 27.3 44 ...
##  $ employrate: num  89.3 90.9 90.2 87.4 89.4 ...
##  $ income    : num  17671 19422 21047 18015 27147 ...
##  $ butcher   : Factor w/ 3 levels "0","1","2 or +": 1 1 2 1 3 1 1 1 3 1 ...
##  $ baker     : Factor w/ 3 levels "0","1","2 or +": 3 1 3 2 3 2 1 1 3 2 ...
##  $ postoffice: Factor w/ 2 levels "0","1 or +": 2 1 2 2 2 1 1 1 2 1 ...
##  $ dentist   : Factor w/ 3 levels "0","1 to 2","3 or +": 1 1 3 2 3 1 1 1 3 1 ...
##  $ grocery   : Factor w/ 2 levels "0","1 or +": 1 2 2 2 2 2 2 2 2 1 ...
```

```
##  $ nursery   : Factor w/ 2 levels "0","1 or +": 1 1 2 1 1 1 1 1 2 1 ...
##  $ doctor    : Factor w/ 3 levels "0","1 to 2","3 or +": 1 3 3 3 3 1 1 1 3 1 ...
##  $ chemist   : Factor w/ 3 levels "0","1","2 or +": 2 1 3 2 3 1 1 1 3 1 ...
##  $ restaurant: Factor w/ 4 levels "0","1","2","3 or +": 2 2 4 4 4 3 3 1 4 3 ...
```

```r
summary(mix_data.na)
```

```
##     farmers           tradesmen         managers          workers     
##  Min.   : 0.0000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00  
##  1st Qu.: 0.5125   1st Qu.: 2.772   1st Qu.: 2.795   1st Qu.:28.57  
##  Median : 1.9700   Median : 3.995   Median : 4.650   Median :33.66  
##  Mean   : 3.4650   Mean   : 4.189   Mean   : 5.287   Mean   :33.52  
##  3rd Qu.: 4.6875   3rd Qu.: 5.300   3rd Qu.: 7.147   3rd Qu.:38.40  
##  Max.   :33.3300   Max.   :16.130   Max.   :22.730   Max.   :57.14  
##                                                                     
##    unemployed      middleempl        retired        employrate    
##  Min.   : 0.00   Min.   : 0.000   Min.   : 9.33   Min.   : 75.08  
##  1st Qu.:11.22   1st Qu.: 8.523   1st Qu.:23.25   1st Qu.: 88.35  
##  Median :13.55   Median :11.875   Median :27.45   Median : 90.66  
##  Mean   :13.38   Mean   :11.993   Mean   :28.17   Mean   : 90.30  
##  3rd Qu.:15.59   3rd Qu.:15.440   3rd Qu.:32.14   3rd Qu.: 92.71  
##  Max.   :33.33   Max.   :31.580   Max.   :51.28   Max.   :100.00  
##                                                                   
##      income        butcher       baker       postoffice    dentist   
##  Min.   :12187   0     :371   0     :291   0     :346   0     :380  
##  1st Qu.:18367   1     : 95   1     :128   1 or +:196   1 to 2: 90  
##  Median :19990   2 or +: 76   2 or +:123                3 or +: 72  
##  Mean   :21003                                                      
##  3rd Qu.:22768                                                      
##  Max.   :70062                                                      
##  NA's   :2                                                          
##    grocery      nursery       doctor      chemist     restaurant 
##  0     :365   0     :520   0     :326   0     :357   0     :247  
##  1 or +:177   1 or +: 22   1 to 2: 92   1     :107   1     :122  
##                            3 or +:124   2 or +: 78   2     : 52  
##                                                      3 or +:121  
##                                                                  
##                                                                  
## 
```

```r
dim(mix_data.na)
```

```
## [1] 542  18
```

Eliminación de valores perdidos

```r
mix_data<-na.omit(mix_data.na)
summary(mix_data)
```

```
##     farmers           tradesmen         managers          workers     
##  Min.   : 0.0000   Min.   : 0.000   Min.   : 0.000   Min.   : 7.69  
##  1st Qu.: 0.5025   1st Qu.: 2.780   1st Qu.: 2.825   1st Qu.:28.64  
##  Median : 1.9550   Median : 4.000   Median : 4.650   Median :33.67  
##  Mean   : 3.3544   Mean   : 4.204   Mean   : 5.286   Mean   :33.65  
```

```
##  3rd Qu.: 4.6125   3rd Qu.: 5.312   3rd Qu.: 7.143   3rd Qu.:38.41
##  Max.   :29.0300   Max.   :16.130   Max.   :22.730   Max.   :57.14
##    unemployed      middleempl        retired        employrate
##  Min.   : 0.00   Min.   : 0.000   Min.   : 9.33   Min.   : 75.08
##  1st Qu.:11.23   1st Qu.: 8.547   1st Qu.:23.23   1st Qu.: 88.35
##  Median :13.55   Median :11.905   Median :27.45   Median : 90.66
##  Mean   :13.35   Mean   :12.005   Mean   :28.16   Mean   : 90.31
##  3rd Qu.:15.55   3rd Qu.:15.465   3rd Qu.:32.14   3rd Qu.: 92.70
##  Max.   :29.19   Max.   :31.580   Max.   :51.28   Max.   :100.00
##      income        butcher       baker      postoffice    dentist
##  Min.   :12187   0     :369   0     :289   0     :344   0     :378
##  1st Qu.:18367   1     : 95   1     :128   1 or +:196   1 to 2: 90
##  Median :19990   2 or +: 76   2 or +:123                3 or +: 72
##  Mean   :21003
##  3rd Qu.:22768
##  Max.   :70062
##    grocery      nursery       doctor      chemist     restaurant
##  0     :363   0     :518   0     :324   0     :355   0     :245
##  1 or +:177   1 or +: 22   1 to 2: 92   1     :107   1     :122
##                            3 or +:124   2 or +: 78   2     : 52
##                                                      3 or +:121
##
##
```

```
dim(mix_data)
```

```
## [1] 540  18
```

2.2.2 Análisis de componentes principales para datos mixtos con PCAmix

División en variables cualitativas y cuantitativas

Construccion de ambos conjuntos de datos: variables cuantitativas (mix_data_quan) y culitativas (mix_data_qual)

```
split<-splitmix(mix_data)
str(split)
```

```
## List of 3
##  $ X.quanti :'data.frame':   540 obs. of  9 variables:
##   ..$ farmers   : num [1:540] 1.98 5.23 0.1 0.18 0.3 ...
##   ..$ tradesmen : num [1:540] 3.68 5.23 4.38 2.29 3.8 5.63 4.21 1.75 4.61 2.3 ...
##   ..$ managers  : num [1:540] 3.97 1.96 5.56 3.7 8.19 1.25 4.21 3.51 5.8 0 ...
##   ..$ workers   : num [1:540] 38.2 21.6 36 42.4 18.6 ...
##   ..$ unemployed: num [1:540] 13.6 15 18.2 15.1 13 ...
##   ..$ middleempl: num [1:540] 9.63 14.38 15.48 8.98 12.07 ...
##   ..$ retired   : num [1:540] 28.9 36.6 20.3 27.3 44 ...
##   ..$ employrate: num [1:540] 89.3 90.9 90.2 87.4 89.4 ...
##   ..$ income    : num [1:540] 17671 19422 21047 18015 27147 ...
##  $ X.quali  :'data.frame':   540 obs. of  9 variables:
##   ..$ butcher   : Factor w/ 3 levels "0","1","2 or +": 1 1 2 1 3 1 1 1 3 1 ...
##   ..$ baker     : Factor w/ 3 levels "0","1","2 or +": 3 1 3 2 3 2 1 1 3 2 ...
##   ..$ postoffice: Factor w/ 2 levels "0","1 or +": 2 1 2 2 2 1 1 1 2 1 ...
##   ..$ dentist   : Factor w/ 3 levels "0","1 to 2","3 or +": 1 1 3 2 3 1 1 1 3 1 ...
##   ..$ grocery   : Factor w/ 2 levels "0","1 or +": 1 2 2 2 2 2 2 2 2 1 ...
```

```
##   ..$ nursery   : Factor w/ 2 levels "0","1 or +": 1 1 2 1 1 1 1 1 2 1 ...
##   ..$ doctor    : Factor w/ 3 levels "0","1 to 2","3 or +": 1 3 3 3 3 1 1 1 3 1 ...
##   ..$ chemist   : Factor w/ 3 levels "0","1","2 or +": 2 1 3 2 3 1 1 1 3 1 ...
##   ..$ restaurant: Factor w/ 4 levels "0","1","2","3 or +": 2 2 4 4 4 3 3 1 4 3 ...
##  $ typ.group: chr "MIX"
```

```r
mix_data_quan<-split$X.quanti
mix_data_qual<-split$X.quali
```

Se aplica PCAmix

No tipifico ni convierto las variables categóricas porque PCAmix ya lo preprocesa

```r
res.pcamix<-PCAmix(X.quanti=mix_data_quan,
                   X.quali=mix_data_qual,
                   rename.level=TRUE,
                   graph=FALSE)

summary(res.pcamix)
```

```
##
## Call:
## PCAmix(X.quanti = mix_data_quan, X.quali = mix_data_qual, rename.level = TRUE,     graph = FALSE)
##
## Method = Factor Analysis of mixed data (FAmix)
##
## Data:
##    number of observations:  540
##    number of  variables:  18
##         number of numerical variables:  9
##         number of categorical variables:  9
##
## Squared loadings :
##            dim1 dim2 dim3 dim4 dim5
## farmers    0.22 0.02 0.01 0.09 0.07
## tradesmen  0.01 0.00 0.00 0.07 0.05
## managers   0.11 0.10 0.37 0.02 0.02
## workers    0.03 0.12 0.01 0.58 0.16
## unemployed 0.09 0.00 0.00 0.06 0.25
## middleempl 0.07 0.01 0.41 0.02 0.01
## retired    0.00 0.00 0.32 0.47 0.01
## employrate 0.06 0.04 0.48 0.02 0.06
## income     0.05 0.08 0.46 0.03 0.00
## butcher    0.62 0.13 0.03 0.00 0.01
## baker      0.76 0.35 0.01 0.00 0.08
## postoffice 0.67 0.08 0.00 0.00 0.01
## dentist    0.81 0.39 0.04 0.04 0.06
## grocery    0.19 0.01 0.04 0.01 0.06
## nursery    0.23 0.15 0.01 0.04 0.03
## doctor     0.84 0.42 0.02 0.01 0.04
## chemist    0.87 0.52 0.07 0.06 0.01
## restaurant 0.68 0.26 0.05 0.02 0.28
```

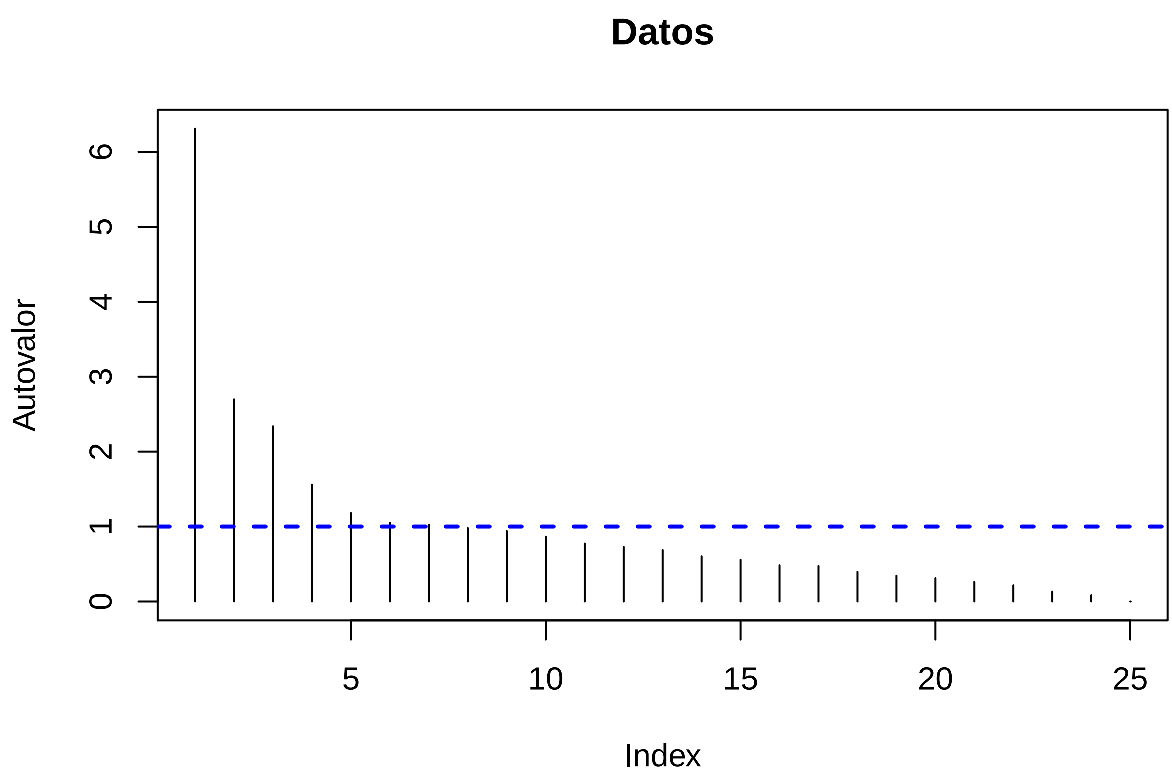
Autovalores

```
round(res.pcamix$eig, 3)
```

```
##        Eigenvalue Proportion Cumulative
## dim 1       6.310     25.241     25.241
## dim 2       2.697     10.789     36.030
## dim 3       2.338      9.351     45.381
## dim 4       1.560      6.241     51.622
## dim 5       1.180      4.719     56.341
## dim 6       1.051      4.203     60.544
## dim 7       1.024      4.097     64.641
## dim 8       0.979      3.917     68.558
## dim 9       0.939      3.757     72.315
## dim 10      0.866      3.464     75.779
## dim 11      0.773      3.094     78.872
## dim 12      0.729      2.916     81.788
## dim 13      0.687      2.749     84.537
## dim 14      0.603      2.412     86.949
## dim 15      0.558      2.231     89.180
## dim 16      0.484      1.934     91.115
## dim 17      0.475      1.900     93.015
## dim 18      0.397      1.587     94.601
## dim 19      0.345      1.378     95.980
## dim 20      0.311      1.244     97.224
## dim 21      0.262      1.049     98.273
## dim 22      0.216      0.865     99.138
## dim 23      0.132      0.528     99.667
## dim 24      0.083      0.333    100.000
## dim 25      0.000      0.000    100.000
```

Gráfico de sedimentación

```
plot(res.pcamix$eig[,1], type="h", main="Datos", ylab="Autovalor")
abline(h=mean(res.pcamix$eig[,1]), lwd=2, lty=2, col="blue")
```



Tomo las 8 primeras componentes que son las que tienen autovalor > 1. Estas 8 componentes explican un *68.6%* de la varianza total.

Por defecto PCAmix muestra las 5 primeras componentes, utilizo el parámetro *ndim* para que muestre 8.

```
res.pcamix8<-PCAmix(X.quanti=mix_data_quan,
                   X.quali=mix_data_qual,
                   rename.level=TRUE,
                   ndim=8,
                   graph=FALSE)

summary(res.pcamix8)
```

```
##
## Call:
## PCAmix(X.quanti = mix_data_quan, X.quali = mix_data_qual, ndim = 8,     rename.level = TRUE, graph =
##
## Method = Factor Analysis of mixed data (FAmix)
##
## Data:
##    number of observations:  540
##    number of  variables:  18
##         number of numerical variables:  9
##         number of categorical variables:  9
##
## Squared loadings :
##              dim1 dim2 dim3 dim4 dim5 dim6 dim7 dim8
## farmers      0.22 0.02 0.01 0.09 0.07 0.13 0.00 0.01
```

```
## tradesmen  0.01 0.00 0.00 0.07 0.05 0.52 0.04 0.13
## managers   0.11 0.10 0.37 0.02 0.02 0.03 0.00 0.00
## workers    0.03 0.12 0.01 0.58 0.16 0.03 0.00 0.01
## unemployed 0.09 0.00 0.00 0.06 0.25 0.13 0.12 0.28
## middleempl 0.07 0.01 0.41 0.02 0.01 0.01 0.00 0.02
## retired    0.00 0.00 0.32 0.47 0.01 0.00 0.00 0.04
## employrate 0.06 0.04 0.48 0.02 0.06 0.01 0.01 0.00
## income     0.05 0.08 0.46 0.03 0.00 0.00 0.01 0.00
## butcher    0.62 0.13 0.03 0.00 0.01 0.00 0.17 0.13
## baker      0.76 0.35 0.01 0.00 0.08 0.04 0.05 0.01
## postoffice 0.67 0.08 0.00 0.00 0.01 0.00 0.00 0.00
## dentist    0.81 0.39 0.04 0.04 0.06 0.00 0.08 0.09
## grocery    0.19 0.01 0.04 0.01 0.06 0.01 0.21 0.00
## nursery    0.23 0.15 0.01 0.04 0.03 0.03 0.05 0.01
## doctor     0.84 0.42 0.02 0.01 0.04 0.02 0.03 0.00
## chemist    0.87 0.52 0.07 0.06 0.01 0.01 0.00 0.00
## restaurant 0.68 0.26 0.05 0.02 0.28 0.09 0.23 0.24
```

Inercia total

```r
# Inercia total p1+m-p2
# p1: numero de variables cuantitativas
# p2: numero de variables cualitativas
# m: numero total de categorias de todas las variables categóricas
sum(res.pcamix8$eig[,1])
```

```
## [1] 25
```

Squared loading

A continuación mostraremos los valores de *squared loading* de cada variable, que es la contribución de esta variable a cada compomente. Es decir, la parte de varianza de la componete considerada explicada por la variable.

```r
round(res.pcamix8$sqload, 3)
```

```
##             dim1  dim2  dim3  dim4  dim5  dim6  dim7  dim8
## farmers    0.221 0.021 0.007 0.092 0.068 0.127 0.001 0.011
## tradesmen  0.009 0.002 0.001 0.073 0.052 0.518 0.044 0.131
## managers   0.107 0.097 0.366 0.022 0.015 0.026 0.000 0.002
## workers    0.028 0.121 0.009 0.576 0.157 0.034 0.003 0.007
## unemployed 0.087 0.005 0.003 0.064 0.245 0.128 0.119 0.285
## middleempl 0.070 0.013 0.411 0.021 0.007 0.009 0.001 0.019
## retired    0.001 0.000 0.324 0.471 0.008 0.000 0.001 0.041
## employrate 0.056 0.044 0.476 0.017 0.060 0.005 0.011 0.000
## income     0.049 0.078 0.462 0.030 0.000 0.003 0.006 0.003
## butcher    0.622 0.133 0.032 0.001 0.013 0.000 0.173 0.126
## baker      0.765 0.352 0.011 0.003 0.079 0.043 0.054 0.008
## postoffice 0.668 0.079 0.000 0.001 0.006 0.000 0.002 0.002
## dentist    0.807 0.388 0.045 0.037 0.055 0.002 0.081 0.092
## grocery    0.188 0.012 0.043 0.012 0.061 0.009 0.210 0.004
## nursery    0.232 0.148 0.005 0.037 0.026 0.027 0.051 0.012
## doctor     0.844 0.425 0.022 0.014 0.037 0.020 0.035 0.003
```

```
## chemist    0.874 0.521 0.074 0.064 0.014 0.007 0.004 0.000
## restaurant 0.683 0.257 0.047 0.023 0.276 0.091 0.230 0.235
```

Para cada variable cuantitativa la suma de las squared loadings de cada componente suman 1. Para las variables cualitativas la suma corresponderá al número de categorías diferentes a 0. Por tanto, si sumamos las filas de la matriz anterior otendremos un valor algo menor al esperado porque solo hemos tomado 8 componentes.

```
apply(res.pcamix8$sqload, 1, sum)
```

```
##    farmers  tradesmen   managers    workers unemployed middleempl
##  0.5484294  0.8293933  0.6367152  0.9346716  0.9360779  0.5507011
##    retired employrate     income    butcher      baker postoffice
##  0.8474407  0.6698572  0.6302990  1.0992651  1.3144951  0.7575675
##    dentist    grocery    nursery     doctor    chemist restaurant
##  1.5077908  0.5377184  0.5391854  1.3991078  1.5572599  1.8434939
```

Veamos que el resultado cuando tomamos las 25 componentes:

```
res.pcamix25<-PCAmix(X.quanti=mix_data_quan,
                     X.quali=mix_data_qual,
                     rename.level=TRUE,
                     ndim=25,
                     graph=FALSE)


apply(res.pcamix25$sqload, 1, sum)
```

```
##    farmers  tradesmen   managers    workers unemployed middleempl
##          1          1          1          1          1          1
##    retired employrate     income    butcher      baker postoffice
##          1          1          1          2          2          1
##    dentist    grocery    nursery     doctor    chemist restaurant
##          2          1          1          2          2          3
```

Contribuciones relativas

La inercia total se reparte entre las distintas dimensiones, permite determinar el nivel de realación entre cada variable y cada componente. A continuación calcularemos las contribuciones relativas para las variables cualitativas y cuantitativas

```
A=rbind(100*res.pcamix8$quali$contrib.pct, # Contribuciones relativas de las cualitativas
        res.pcamix8$quanti$contrib.pct)     # Contribuciones porcentuales de las cuantitativas

round(A, 3)
```

```
##              dim1   dim2  dim3  dim4   dim5  dim6   dim7   dim8
## butcher     9.857  4.948 1.368 0.066  1.061 0.015 16.848 12.820
## baker      12.117 13.033 0.465 0.196  6.730 4.120  5.268  0.793
## postoffice 10.583  2.941 0.003 0.033  0.506 0.007  0.222  0.158
## dentist    12.788 14.388 1.911 2.391  4.681 0.222  7.891  9.436
## grocery     2.985  0.458 1.825 0.744  5.132 0.826 20.495  0.365
## nursery     3.674  5.497 0.225 2.386  2.205 2.611  4.985  1.238
## doctor     13.370 15.753 0.933 0.918  3.108 1.929  3.409  0.259
## chemist    13.847 19.306 3.145 4.097  1.200 0.654  0.415  0.002
## restaurant 10.819  9.539 2.027 1.496 23.415 8.692 22.429 24.045
```

```
## farmers     3.508  0.791  0.317  5.909  5.733 12.110  0.057  1.091
## tradesmen   0.136  0.082  0.040  4.699  4.408 49.292  4.264 13.350
## managers    1.703  3.599 15.653  1.429  1.274  2.510  0.021  0.235
## workers     0.445  4.489  0.377 36.946 13.348  3.193  0.247  0.688
## unemployed  1.374  0.176  0.127  4.110 20.784 12.200 11.663 29.072
## middleempl  1.115  0.480 17.560  1.361  0.584  0.861  0.081  1.925
## retired     0.014  0.015 13.875 30.186  0.709  0.003  0.094  4.232
## employrate  0.889  1.621 20.376  1.080  5.119  0.494  1.069  0.026
## income      0.775  2.883 19.771  1.952  0.004  0.264  0.543  0.265
```

Comprobamos que la suma para cada columna es igual a 100

```r
apply(A,2,sum)
```

```
## dim1 dim2 dim3 dim4 dim5 dim6 dim7 dim8
##  100  100  100  100  100  100  100  100
```

Coordenadas

A continuación mostraremos las coordenadas de cada dimensión

```r
head(res.pcamix8$ind$coord)
```

```
##                         dim 1      dim 2      dim 3      dim 4       dim 5
## ABZAC               0.3089595 -1.3275558 -0.3797857 -0.3256275  0.08540901
## AILLAS             -0.5151541  0.4860533 -0.6130975  1.1286307  1.83588594
## AMBARES-ET-LAGRAVE  5.4067580  2.1560126 -0.5016042 -2.2518391  0.77006102
## AMBES               2.4031163 -2.7811727 -0.8398837 -0.6566105 -0.55942986
## ANDERNOS-LES-BAINS  5.0613694  2.5346005 -1.1770582  2.2782578  0.24678929
## ANGLADE            -1.1175075 -1.6720510 -1.8199727 -0.9576146  0.43372770
##                         dim 6      dim 7      dim 8
## ABZAC              -0.5682426 -0.9321933 -0.6063410
## AILLAS             -0.3185143  0.3503863 -0.4641625
## AMBARES-ET-LAGRAVE  0.2871217  0.1020615 -1.0092159
## AMBES               0.3000971 -0.6565987  0.5010043
## ANDERNOS-LES-BAINS -0.2846693  0.7910423 -0.6048800
## ANGLADE             0.7310601  2.2561144  1.7331660
```

```r
#Coordenadas de las categ. de las cualitativas:
res.pcamix8$levels$coord
```

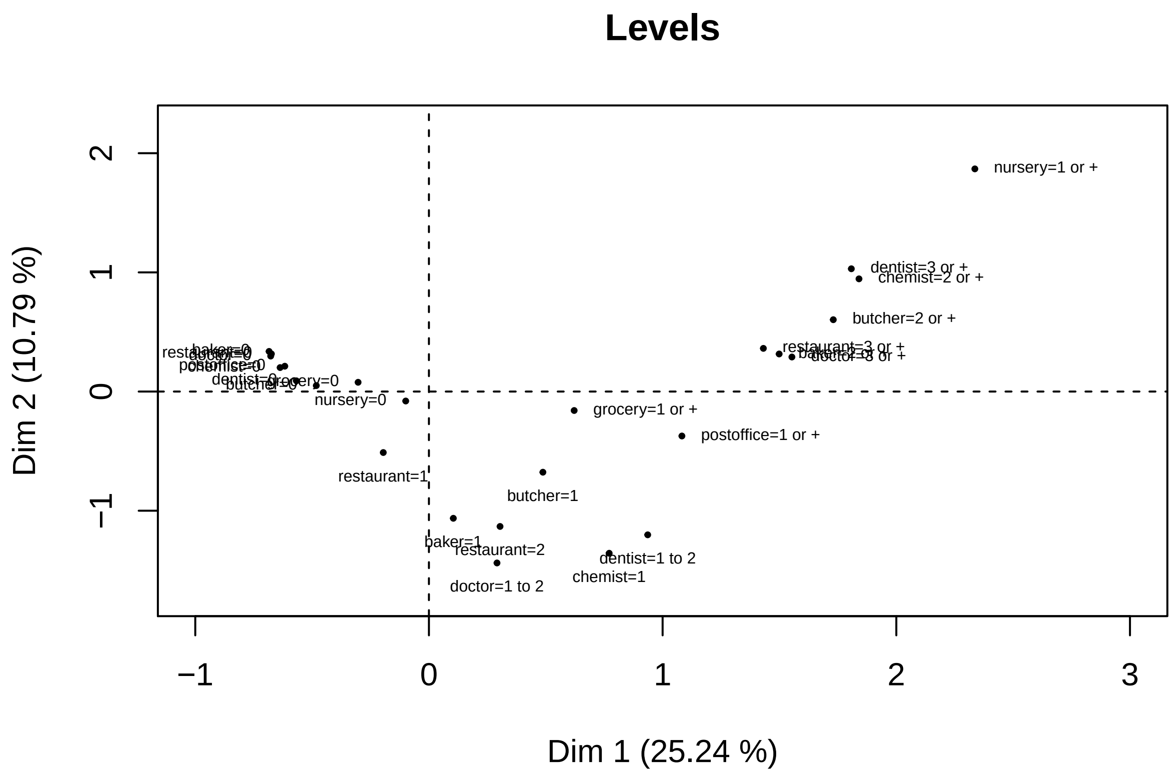
```
##                          dim1        dim2         dim3         dim4
## butcher=0         -0.48196600  0.05013436  0.066880538 -0.016613015
## butcher=1          0.48771411 -0.67694879  0.093225025  0.068987487
## butcher=2 or +     1.73042915  0.60277049 -0.441253891 -0.005573798
## baker=0           -0.68409116  0.33685588 -0.002428494 -0.016954762
## baker=1            0.10443724 -1.06360512  0.152655538 -0.055652622
## baker=2 or +       1.49865347  0.31536672 -0.153155074  0.097751722
## postoffice=0      -0.61683304  0.21259870  0.006487369 -0.017198887
## postoffice=1 or +  1.08260493 -0.37313240 -0.011385994  0.030185802
## dentist=0         -0.56721719  0.09000806 -0.037445514 -0.049304414
## dentist=1 to 2     0.93619965 -1.20223507  0.419618478  0.405444488
## dentist=3 or +     1.80764069  1.03025152 -0.327934149 -0.247957438
## grocery=0         -0.30306382  0.07759064  0.144247186  0.075237212
## grocery=1 or +     0.62153767 -0.15912656 -0.295828976 -0.154300044
## nursery=0         -0.09922205 -0.07935560  0.014945475  0.039761273
```

```
## nursery=1 or +     2.33622832  1.86846358 -0.351898002 -0.936197237
## doctor=0          -0.67719493  0.29753150 -0.056535699 -0.083271011
## doctor=1 to 2      0.29130562 -1.43842819  0.324330998  0.239964955
## doctor=3 or +      1.55331485  0.28979989 -0.092910367  0.039540578
## chemist=0         -0.63686061  0.20122769 -0.019341021 -0.099390321
## chemist=1          0.77136200 -1.35705618  0.430741268  0.500033522
## chemist=2 or +     1.84038181  0.94575875 -0.502862220 -0.233590039
## restaurant=0      -0.67417850  0.31614517 -0.094975088  0.032383701
## restaurant=1      -0.19526948 -0.51168988  0.370659763 -0.268420980
## restaurant=2       0.30430313 -1.13200330  0.100319481  0.148122243
## restaurant=3 or +  1.43118054  0.36227082 -0.224530640  0.141413192
##                          dim5          dim6        dim7          dim8
## butcher=0          0.02419908 -0.0070353978  0.09927658  0.1361529586
## butcher=1          0.11637077  0.0259919833 -0.83246774 -0.7586427537
## butcher=2 or +    -0.26295634  0.0016687286  0.55857077  0.2872449982
## baker=0           -0.04218848 -0.0630536599 -0.05903365  0.0538859846
## baker=1            0.44846074  0.3554951822  0.38928884 -0.1576668113
## baker=2 or +      -0.36756507 -0.2217957366 -0.26640851  0.0374658723
## postoffice=0       0.05831377  0.0065509511  0.03599167  0.0296542742
## postoffice=1 or + -0.10234661 -0.0114975876 -0.06316905 -0.0520462772
## dentist=0          0.11909516  0.0009525385  0.03999813 -0.0667077780
## dentist=1 to 2    -0.52078670 -0.0810739307 -0.54674320  0.6232876065
## dentist=3 or +     0.02573376  0.0963415862  0.47343883 -0.4288936736
## grocery=0         -0.17181586 -0.0650665437 -0.31993656  0.0417569354
## grocery=1 or +     0.35236813  0.1334415558  0.65614108 -0.0856371049
## nursery=0         -0.03324116 -0.0341334243 -0.04656628  0.0226854786
## nursery=1 or +     0.78267831  0.8036869895  1.09642417 -0.5341399049
## doctor=0           0.04784992 -0.0257611144 -0.02176244  0.0126839067
## doctor=1 to 2      0.26285730  0.2934576099  0.37214993 -0.1069661045
## doctor=3 or +     -0.32005037 -0.1504153150 -0.21924808  0.0462201278
## chemist=0          0.08585431 -0.0238665132  0.03146547  0.0029360765
## chemist=1         -0.15894731  0.1566532950 -0.13113767 -0.0092967411
## chemist=2 or +    -0.17270407 -0.1062729535  0.03668576 -0.0006096904
## restaurant=0      -0.22062669  0.1505518427 -0.08562999  0.1938183905
## restaurant=1       0.92123128 -0.3263317748 -0.08558533 -0.7293650020
## restaurant=2      -0.77162055  0.6597844300  1.42887727  1.0010487222
## restaurant=3 or + -0.15051577 -0.2593513661 -0.35438727 -0.0872521405
```

Representacion gráfica

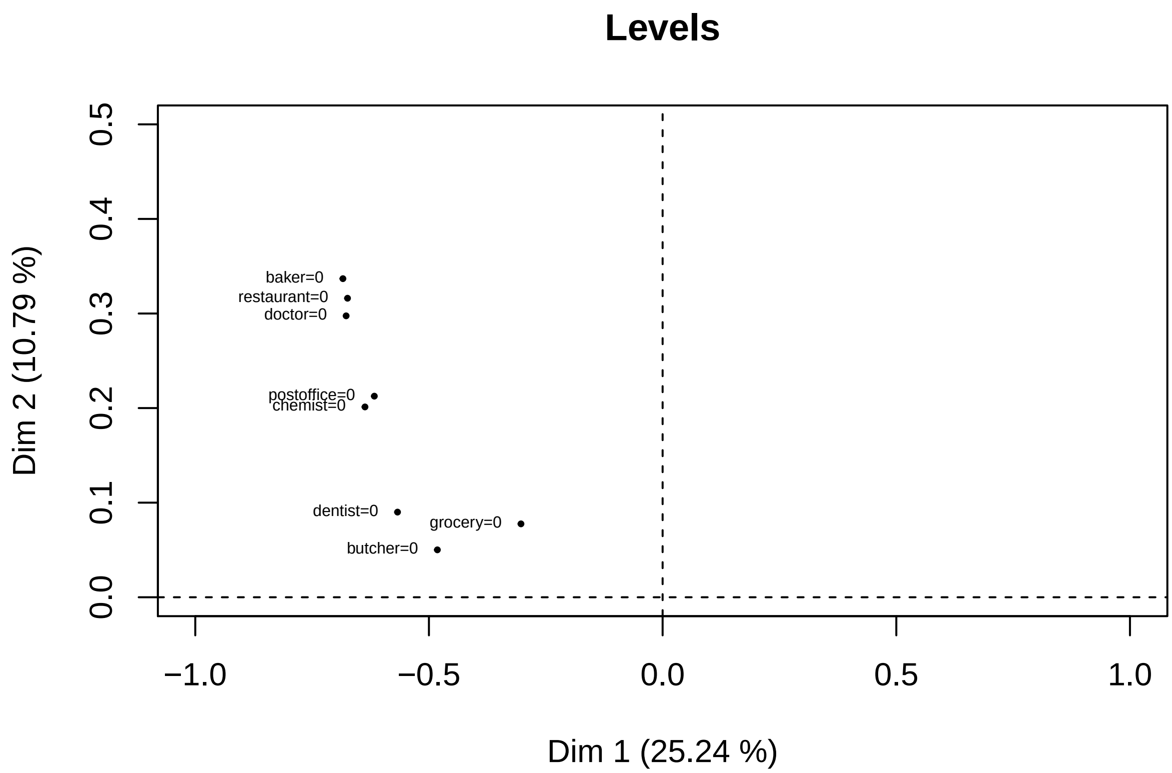
Variables cualitativas

```r
plot(res.pcamix8, choice="levels",
     axes=c(1,2), xlim=c(-1, 3),
     cex=0.5, main="Levels")
```



Ampliamos el primer cuadrante del gráfico anterior

```
plot(res.pcamix8, choice="levels",
     axes=c(1,2), xlim=c(-1, 1), ylim=c(0, 0.5),
     cex=0.5, main="Levels")
```



Se observa que la dimensión 1 separa las ciudades en funcion del número de servicios que ofrezcan. Las ciudades con mayor número de servicios quedan a la derecha (toman valores mayores) mientras que las que ofrecen menor número de servicios quedan a las izquierda.

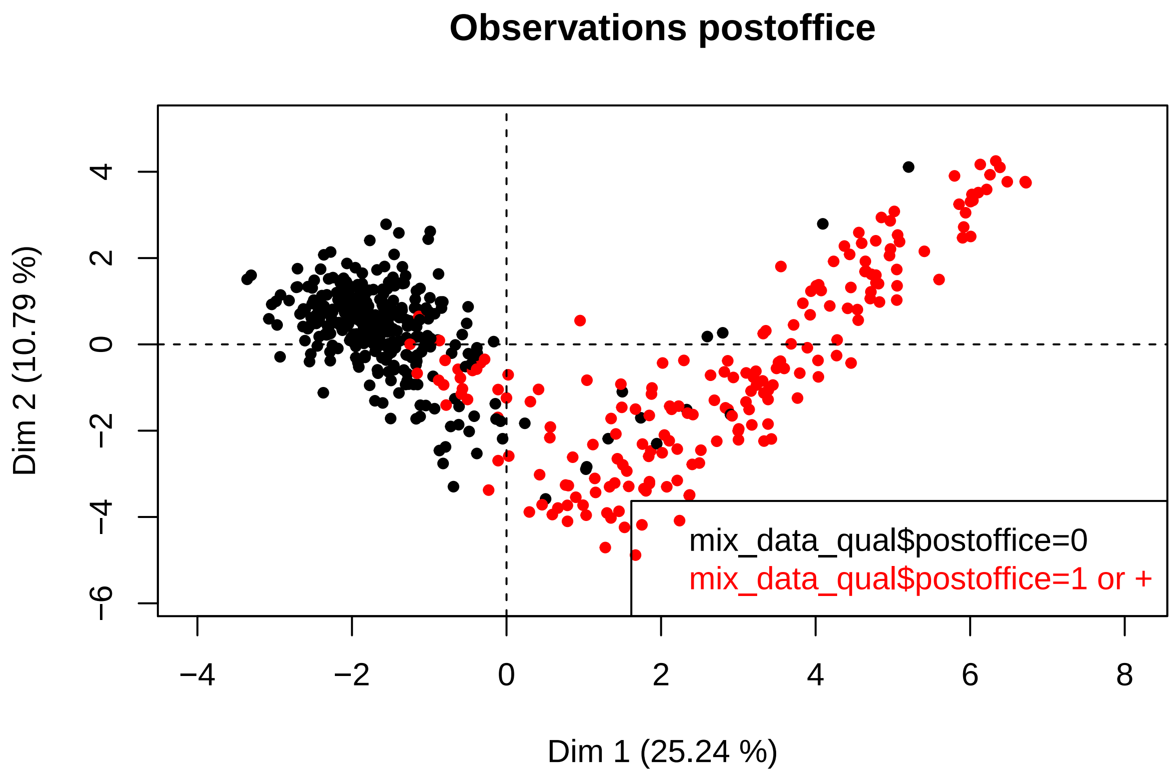
El primer cuadrante (Dim1<0, Dim2>0) es el que presenta menor procentaje de servicios.

El segundo cuadrante (Dim 1>0, Dim 2 >0) es el que presenta el mayor porcentaje de servicios.

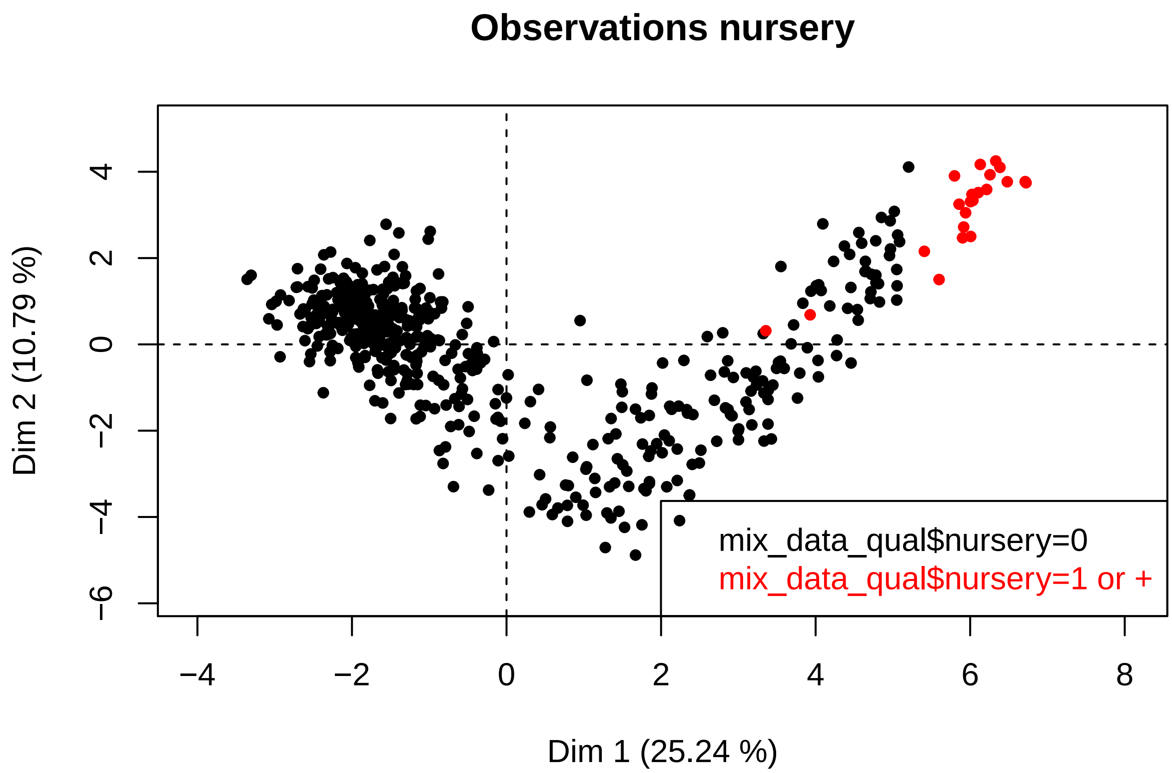
Observaciones

Se representarán algunas de las observaciones

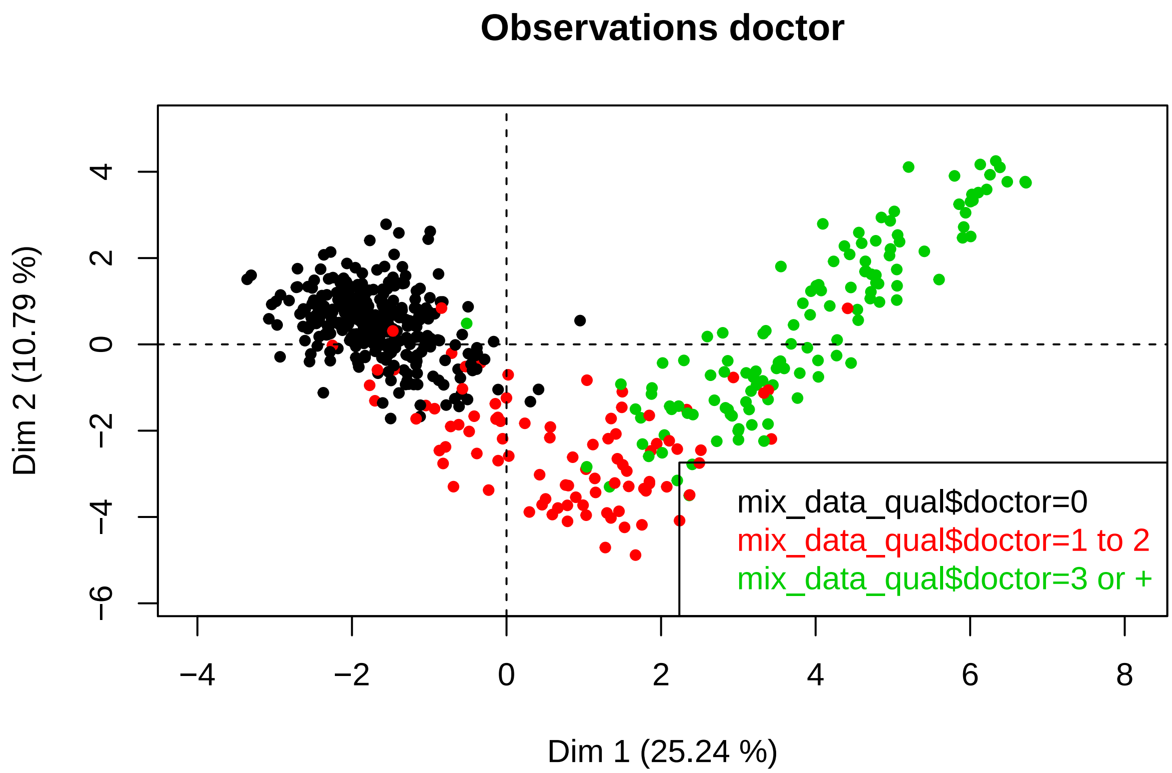
```
plot(res.pcamix8, choice="ind", axes=c(1,2),
     coloring.ind=mix_data_qual$postoffice,
     label=FALSE,
     posleg="bottomright", main="Observations postoffice")
```




```
plot(res.pcamix8, choice="ind", axes=c(1,2),
     coloring.ind=mix_data_qual$nursery,
     label=FALSE,
     posleg="bottomright", main="Observations nursery")
```



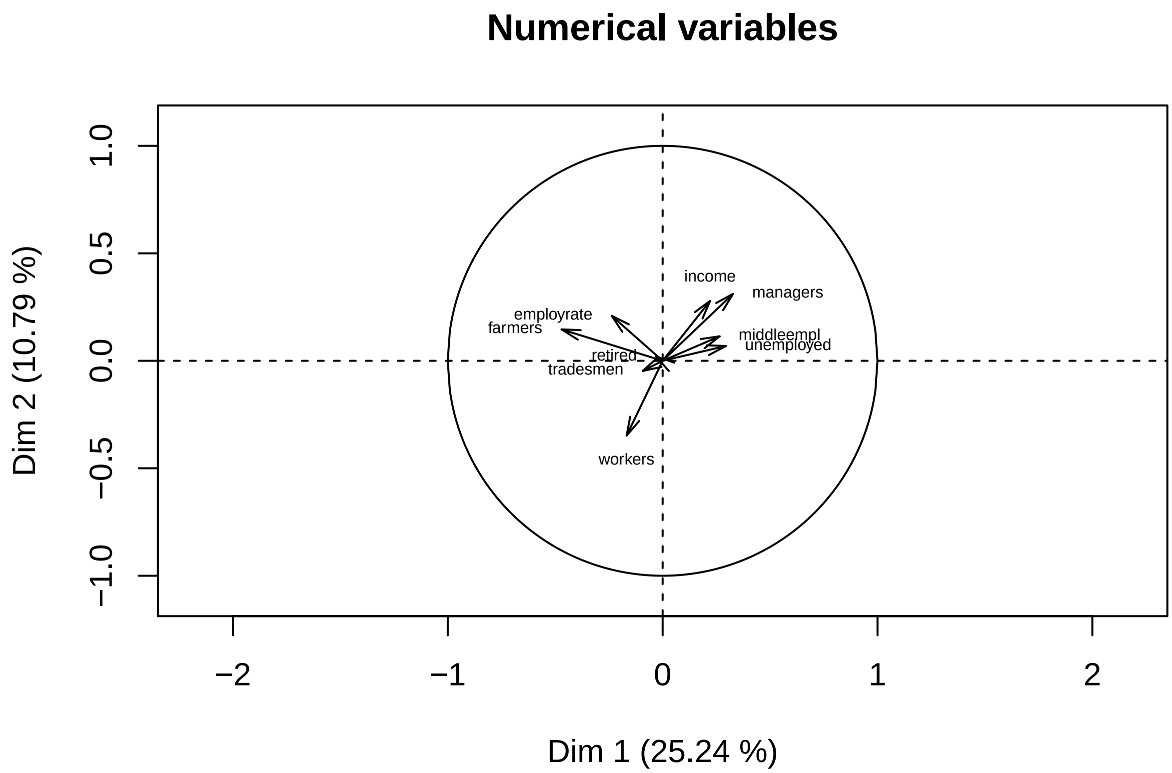

```r
plot(res.pcamix8, choice="ind", axes=c(1,2),
     coloring.ind=mix_data_qual$doctor,
     label=FALSE,
     posleg="bottomright", main="Observations doctor")
```



Se observa un comportamiento similar al mencionado anteriormente. Las ciudades con mayor porcentaje de servicios se encuentran en la parte derecha.

Variables numéricas

```
plot(res.pcamix8, choice="cor", axes=c(1,2),
     main="Numerical variables",
     cex=0.5)
```



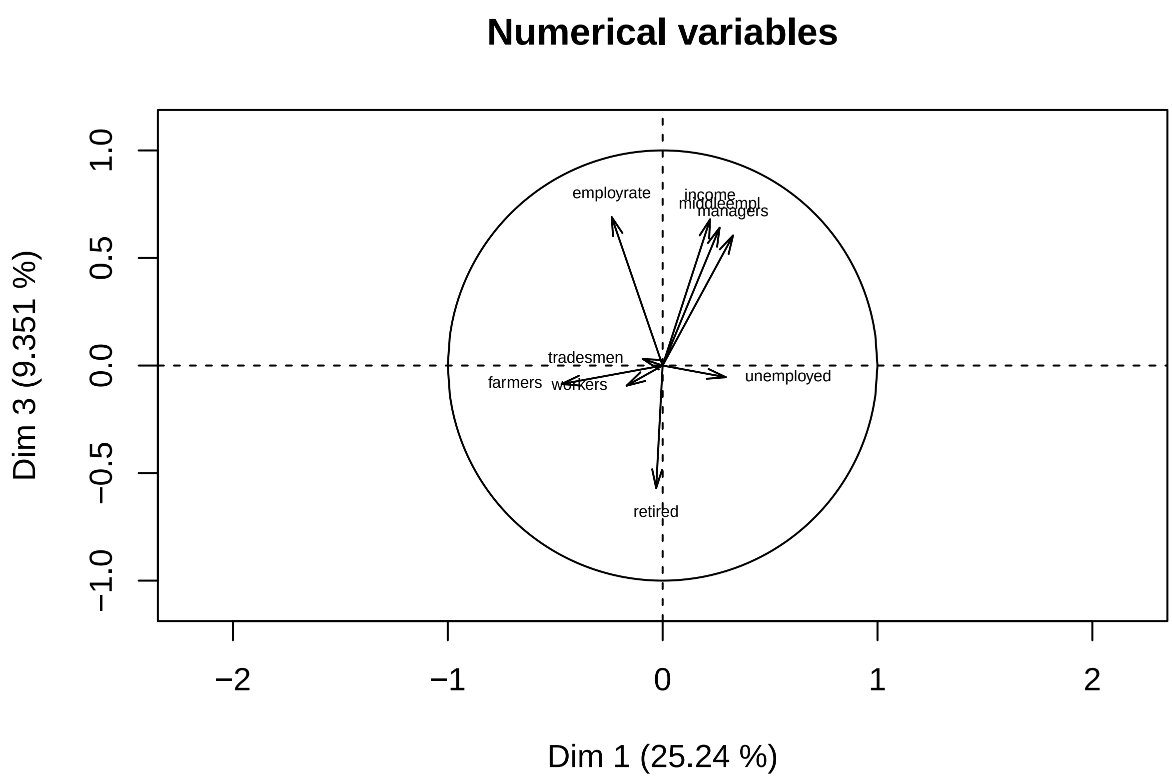
Se observa que el número de trabajadores (workers) está inversamente correlado con el salario medio (income) y con el número de directores (managers).

Tambien se observa que el número de desempleados (unemployed) presenta correlación inversa con el número de profesionales cualificado (tradesmen) y con la tasa de empleo (employrate)

Si relacionamos este gráfico con el anterior observamos que las ciudades donde el salario medio es mayor hay mayor número de servicios.

Para la dimensión 3

```
plot(res.pcamix8, choice="cor", axes=c(1,3),
     main="Numerical variables",
     cex=0.5)
```

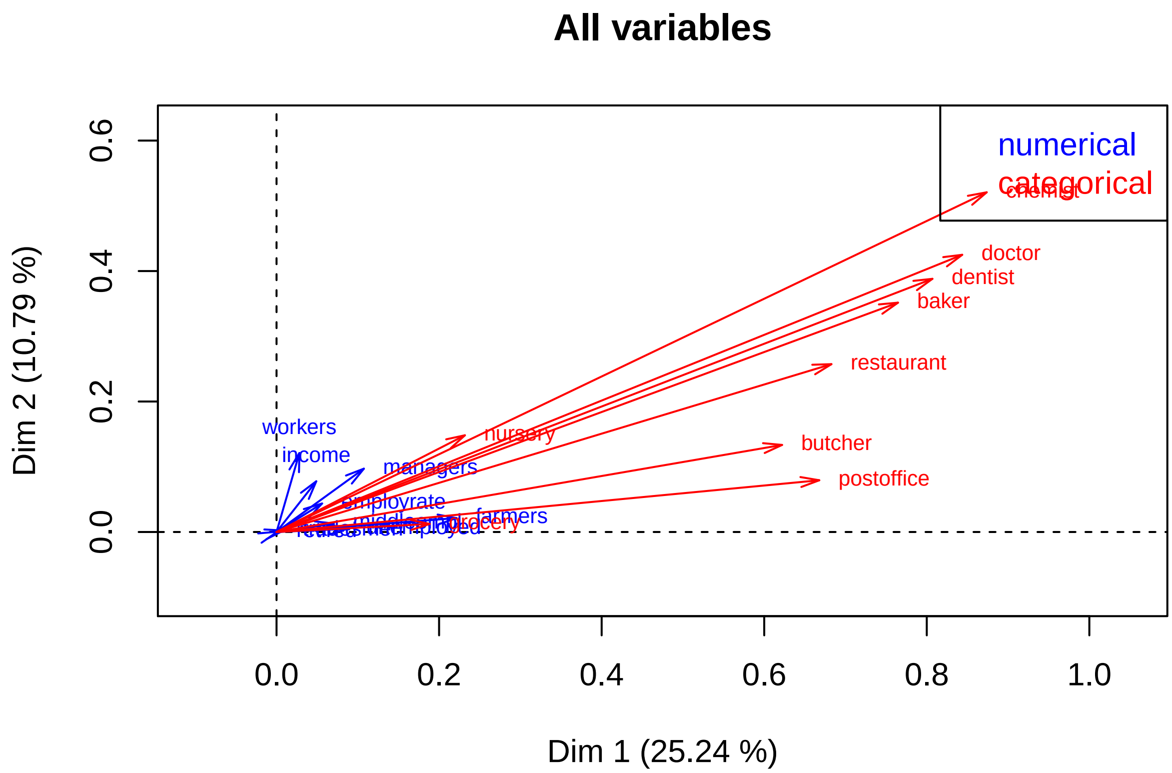


Observamos al representar la dimensión 3 que queda bastante explicada con la variable *retired*.

Todas las variables

Dimensión 1 vs Dimensión 2

```
plot(res.pcamix8, choice="sqload", axes=c(1,2),
     coloring.var="type", leg=TRUE,
     xlim=c(-0.1,1.05),posleg="topright",
     main="All variables",
     cex=0.7)
```

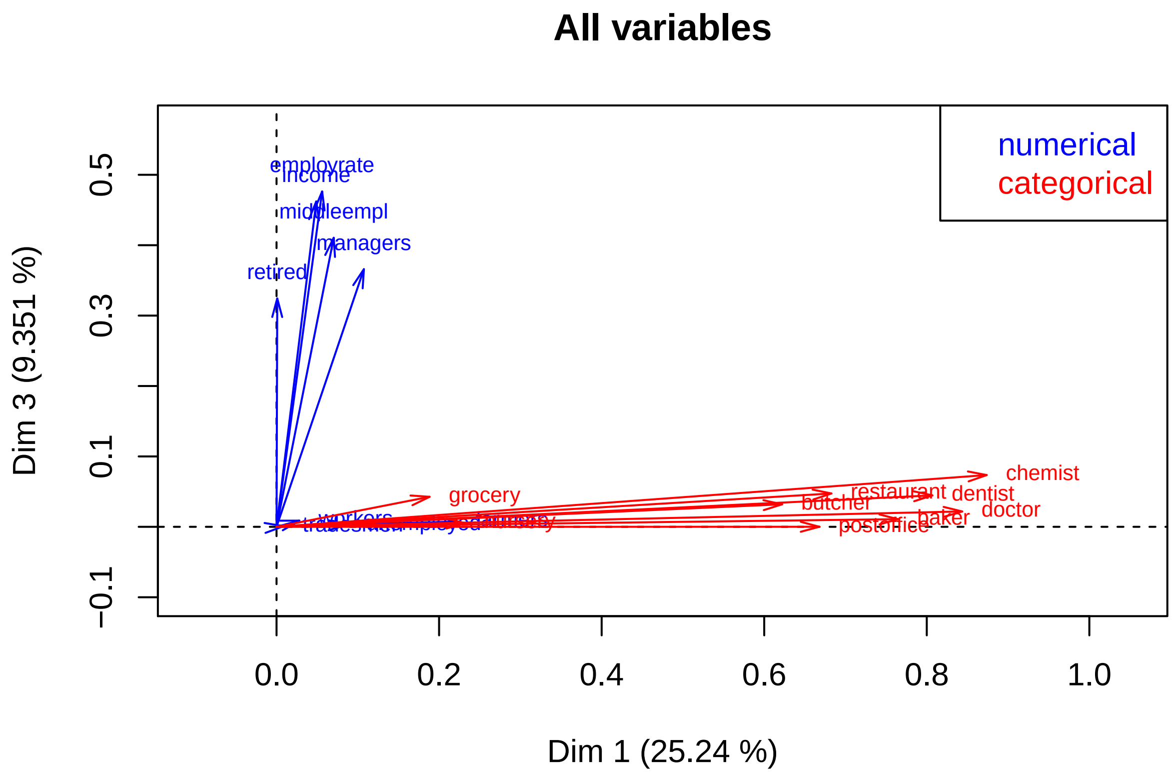



```r
str(mix_data_qual)
```

```
## 'data.frame':    540 obs. of  9 variables:
##  $ butcher   : Factor w/ 3 levels "0","1","2 or +": 1 1 2 1 3 1 1 1 3 1 ...
##  $ baker     : Factor w/ 3 levels "0","1","2 or +": 3 1 3 2 3 2 1 1 3 2 ...
##  $ postoffice: Factor w/ 2 levels "0","1 or +": 2 1 2 2 2 1 1 1 2 1 ...
##  $ dentist   : Factor w/ 3 levels "0","1 to 2","3 or +": 1 1 3 2 3 1 1 1 3 1 ...
##  $ grocery   : Factor w/ 2 levels "0","1 or +": 1 2 2 2 2 2 2 2 2 1 ...
##  $ nursery   : Factor w/ 2 levels "0","1 or +": 1 1 2 1 1 1 1 1 2 1 ...
##  $ doctor    : Factor w/ 3 levels "0","1 to 2","3 or +": 1 3 3 3 3 1 1 1 3 1 ...
##  $ chemist   : Factor w/ 3 levels "0","1","2 or +": 2 1 3 2 3 1 1 1 3 1 ...
##  $ restaurant: Factor w/ 4 levels "0","1","2","3 or +": 2 2 4 4 4 3 3 1 4 3 ...
```

Dimensión 1 vs Dimensión 3

```r
plot(res.pcamix8, choice="sqload", axes=c(1,3),
     coloring.var="type", leg=TRUE,
     xlim=c(-0.1,1.05),posleg="topright",
     main="All variables",
     cex=0.7)
```




```r
str(mix_data_qual)
```

```
## 'data.frame':    540 obs. of  9 variables:
##  $ butcher   : Factor w/ 3 levels "0","1","2 or +": 1 1 2 1 3 1 1 1 3 1 ...
##  $ baker     : Factor w/ 3 levels "0","1","2 or +": 3 1 3 2 3 2 1 1 3 2 ...
##  $ postoffice: Factor w/ 2 levels "0","1 or +": 2 1 2 2 2 1 1 1 2 1 ...
##  $ dentist   : Factor w/ 3 levels "0","1 to 2","3 or +": 1 1 3 2 3 1 1 1 3 1 ...
##  $ grocery   : Factor w/ 2 levels "0","1 or +": 1 2 2 2 2 2 2 2 2 1 ...
##  $ nursery   : Factor w/ 2 levels "0","1 or +": 1 1 2 1 1 1 1 1 2 1 ...
##  $ doctor    : Factor w/ 3 levels "0","1 to 2","3 or +": 1 3 3 3 3 1 1 1 3 1 ...
##  $ chemist   : Factor w/ 3 levels "0","1","2 or +": 2 1 3 2 3 1 1 1 3 1 ...
##  $ restaurant: Factor w/ 4 levels "0","1","2","3 or +": 2 2 4 4 4 3 3 1 4 3 ...
```

Vemos que la dimensión 3 queda explicada con las variables numéricas, mientras que la dimensión 1 está mejor explicada por las categóricas.

Ejercicio 2.3

Aplicar procedimientos de selección de variables para construir modelos de regresión lineal donde *income* es la variable dependiente, sobre *gironde$employment*

2.3.0 Preparación de los datos

Inspección de los datos

Tomamos el dataset *employment* construido en los apartados anteriores y para el que ya se han eliminado los valores perdidos

```r
# comprobamos que el numero de valores perdidos es igual a 0, todos los registros son completos
sum(is.na(employment))
```

```
## [1] 0
```

```r
head(employment)
```

```
##                    farmers tradesmen managers workers unemployed
## ABZAC                 1.98      3.68     3.97   38.25      13.60
## AILLAS                5.23      5.23     1.96   21.57      15.03
## AMBARES-ET-LAGRAVE    0.10      4.38     5.56   35.98      18.23
## AMBES                 0.18      2.29     3.70   42.42      15.11
## ANDERNOS-LES-BAINS    0.30      3.80     8.19   18.65      13.04
## ANGLADE               3.13      5.63     1.25   39.37      16.87
##                    middleempl retired employrate   income
## ABZAC                    9.63   28.90      89.26 17670.60
## AILLAS                  14.38   36.60      90.88 19422.49
## AMBARES-ET-LAGRAVE      15.48   20.28      90.25 21047.07
## AMBES                    8.98   27.33      87.38 18014.52
## ANDERNOS-LES-BAINS      12.07   43.97      89.43 27147.48
## ANGLADE                  5.63   28.12      88.71 15897.99
```

```r
summary(employment)
```

```
##     farmers          tradesmen         managers         workers
##  Min.   : 0.0000   Min.   : 0.000   Min.   : 0.000   Min.   : 7.69
##  1st Qu.: 0.5025   1st Qu.: 2.780   1st Qu.: 2.825   1st Qu.:28.64
##  Median : 1.9550   Median : 4.000   Median : 4.650   Median :33.67
##  Mean   : 3.3544   Mean   : 4.204   Mean   : 5.286   Mean   :33.65
##  3rd Qu.: 4.6125   3rd Qu.: 5.312   3rd Qu.: 7.143   3rd Qu.:38.41
##  Max.   :29.0300   Max.   :16.130   Max.   :22.730   Max.   :57.14
##    unemployed      middleempl        retired         employrate
##  Min.   : 0.00   Min.   : 0.000   Min.   : 9.33   Min.   : 75.08
##  1st Qu.:11.23   1st Qu.: 8.547   1st Qu.:23.23   1st Qu.: 88.35
##  Median :13.55   Median :11.905   Median :27.45   Median : 90.66
##  Mean   :13.35   Mean   :12.005   Mean   :28.16   Mean   : 90.31
##  3rd Qu.:15.55   3rd Qu.:15.465   3rd Qu.:32.14   3rd Qu.: 92.70
##  Max.   :29.19   Max.   :31.580   Max.   :51.28   Max.   :100.00
##      income
##  Min.   :12187
##  1st Qu.:18367
##  Median :19990
##  Mean   :21003
##  3rd Qu.:22768
##  Max.   :70062
```

```r
str(employment)
```

```
## 'data.frame':    540 obs. of  9 variables:
##  $ farmers   : num  1.98 5.23 0.1 0.18 0.3 ...
##  $ tradesmen : num  3.68 5.23 4.38 2.29 3.8 5.63 4.21 1.75 4.61 2.3 ...
##  $ managers  : num  3.97 1.96 5.56 3.7 8.19 1.25 4.21 3.51 5.8 0 ...
##  $ workers   : num  38.2 21.6 36 42.4 18.6 ...
##  $ unemployed: num  13.6 15 18.2 15.1 13 ...
```

```
##  $ middleempl: num  9.63 14.38 15.48 8.98 12.07 ...
##  $ retired   : num  28.9 36.6 20.3 27.3 44 ...
##  $ employrate: num  89.3 90.9 90.2 87.4 89.4 ...
##  $ income    : num  17671 19422 21047 18015 27147 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:2] 63 369
##   .. ..- attr(*, "names")= chr [1:2] "BOSSUGAN" "SAINT-AVIT-DE-SOULEGE"
```

Comprobamos que no contiene ninguna variable categórica, son todas numéricas, por tanto no hay que realizar ninguna conversión, ya que el algoritmo genético con la librería *GA* necesita que las variables del conjunto de datos de entrada sean numéricas.-

Partición en entrenamiento y test

Para poder comparar los modelos que vamos a construir necesitamos dividir los datos en conjunto test y conjunto de entrenamiento, asi conseguiremos capacidad de generalización comparando R2 y error cometido en los datos test. Destinaremos el 75% a entrenamiento y reservaremos el 25% para test

```
set.seed(123456789)
n=nrow(employment)
indices=1:n
index_train=sample(indices, floor(0.75*n))
index_test<- setdiff(indices, index_train)

employ_train=employment[index_train,]
employ_test=employment[index_test,]
```

A continuación se construirán 3 modelos lineales diferentes, uno sin selección de variables para comparar con el resto, y otros dos modelos realizando previamente selección de variables, uno de ellos usando exploración completa con la librería *leaps*, y el otro modelos realizando selección de variables mediante algoritmos genéticos

2.3.1 Modelo de regresion lineal con todas las variables

Utilizamos la función *Ajuste* vista en clase para calcular MSE, RMSE, R2 y R2 ajustado de cada modelo. La función ha sido ligeramente modificada para que tambien calcule el R2 ajustado, ya que estamos comparando modelos distintos con número de variables distintos.

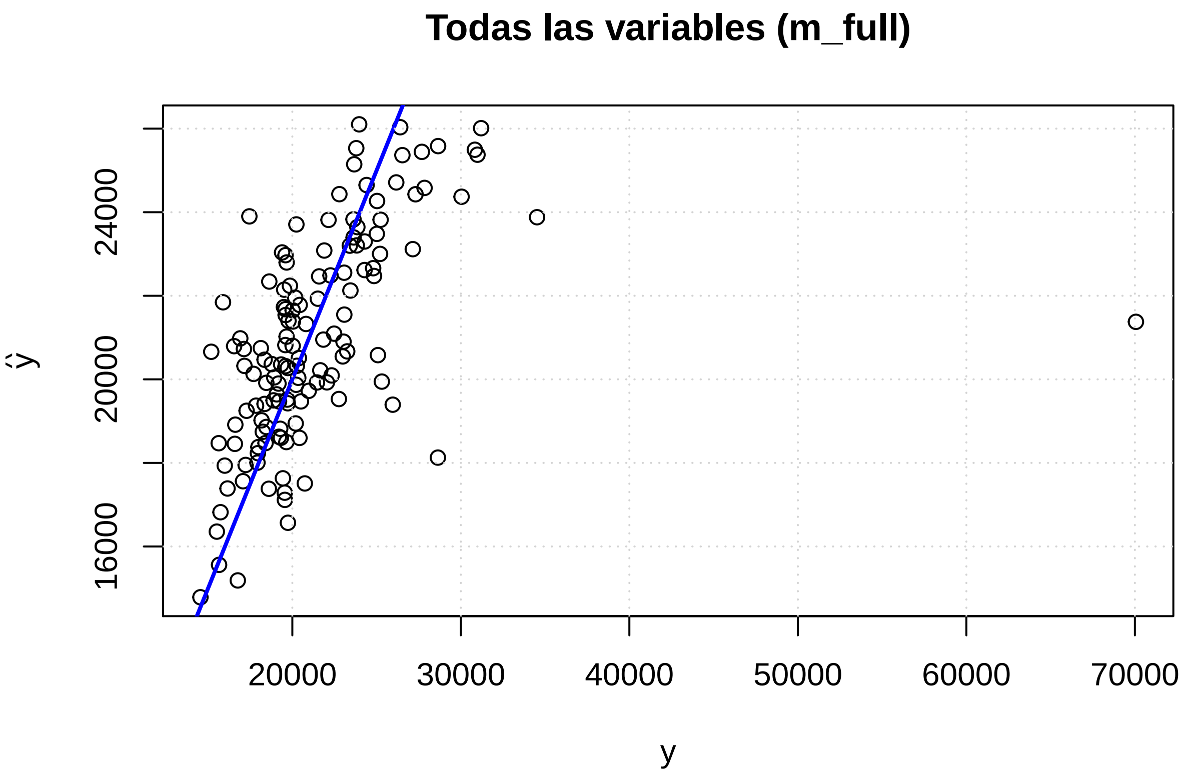
```
Ajuste<- function(y, pred, n, k, titulo)
{
  residuos=y-pred
  plot(y,pred,main=titulo,ylab=expression(hat(y)))
  abline(a=0,b=1,col="blue",lwd=2)
  grid()
  MSE= mean(residuos^2)
  RMSE= sqrt(MSE)
  R2= cor(y,pred)^2
  R2_ajust=1-(n-1)*(1-R2)/(n-k-1)
  return(list(MSE=MSE, RMSE=RMSE, R2=R2, R2_ajust=R2_ajust))
}
```

```
m_full=lm(employ_train$income~.,data=employ_train)
summary(m_full)
```

```
##
## Call:
## lm(formula = employ_train$income ~ ., data = employ_train)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11434.3  -1679.6   -316.5   1501.5  15630.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.415e+06  2.016e+06  -2.686  0.00753 **
## farmers      5.408e+04  2.016e+04   2.683  0.00760 **
## tradesmen    5.416e+04  2.016e+04   2.687  0.00752 **
## managers     5.444e+04  2.016e+04   2.701  0.00722 **
## workers      5.403e+04  2.016e+04   2.680  0.00767 **
## unemployed   5.402e+04  2.016e+04   2.680  0.00767 **
## middleempl   5.428e+04  2.016e+04   2.692  0.00739 **
## retired      5.405e+04  2.016e+04   2.681  0.00765 **
## employrate   2.991e+02  5.407e+01   5.532 5.76e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3160 on 396 degrees of freedom
## Multiple R-squared:  0.4285, Adjusted R-squared:  0.4169
## F-statistic: 37.11 on 8 and 396 DF,  p-value: < 2.2e-16
```

```
pred_full=predict(m_full, employ_test)

# Número de variables independientes en el modelo m_full
k_full=length(employ_test[1,])-1

# Tamaño de la muestra test
n_test=length(employ_test[,1])

(ajuste_full=Ajuste(employ_test$income, pred_full, n_test, k_full, "Todas las variables (m_full)"))
```




```
## $MSE
## [1] 24630697
##
## $RMSE
## [1] 4962.932
##
## $R2
## [1] 0.227129
##
## $R2_ajust
## [1] 0.1780578
```

Se observa que los p-valores son todos >0.05, por tanto todas las variables son significativas. El R2 obtenido es muy bajo, el modelo no se ajusta bien.

2.3.2 Modelo de regresión lineal con selección de variables mediante exploración completa (leaps)

```
exh_search=regsubsets(income~.,data=employ_train, nvmax=13)
(resumen=summary(exh_search))
```

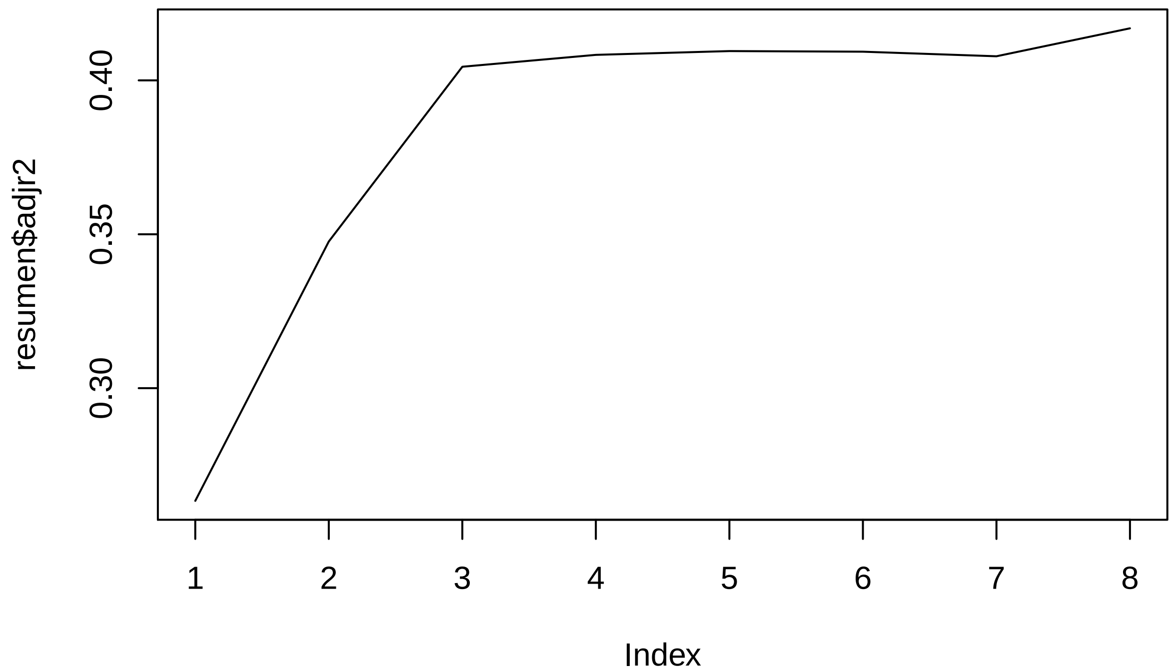
```
## Subset selection object
## Call: regsubsets.formula(income ~ ., data = employ_train, nvmax = 13)
## 8 Variables  (and intercept)
##            Forced in Forced out
```

```
## farmers        FALSE      FALSE
## tradesmen      FALSE      FALSE
## managers       FALSE      FALSE
## workers        FALSE      FALSE
## unemployed     FALSE      FALSE
## middleempl     FALSE      FALSE
## retired        FALSE      FALSE
## employrate     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          farmers tradesmen managers workers unemployed middleempl retired
## 1  ( 1 ) " "     " "       "*"      " "     " "        " "        " "
## 2  ( 1 ) " "     " "       "*"      " "     " "        "*"        " "
## 3  ( 1 ) " "     " "       "*"      " "     " "        "*"        " "
## 4  ( 1 ) " "     "*"       "*"      " "     " "        "*"        " "
## 5  ( 1 ) "*"     "*"       "*"      " "     " "        "*"        " "
## 6  ( 1 ) "*"     "*"       "*"      " "     " "        "*"        "*"
## 7  ( 1 ) "*"     "*"       "*"      " "     "*"        "*"        "*"
## 8  ( 1 ) "*"     "*"       "*"      "*"     "*"        "*"        "*"
##          employrate
## 1  ( 1 ) " "
## 2  ( 1 ) " "
## 3  ( 1 ) "*"
## 4  ( 1 ) "*"
## 5  ( 1 ) "*"
## 6  ( 1 ) "*"
## 7  ( 1 ) "*"
## 8  ( 1 ) "*"
```

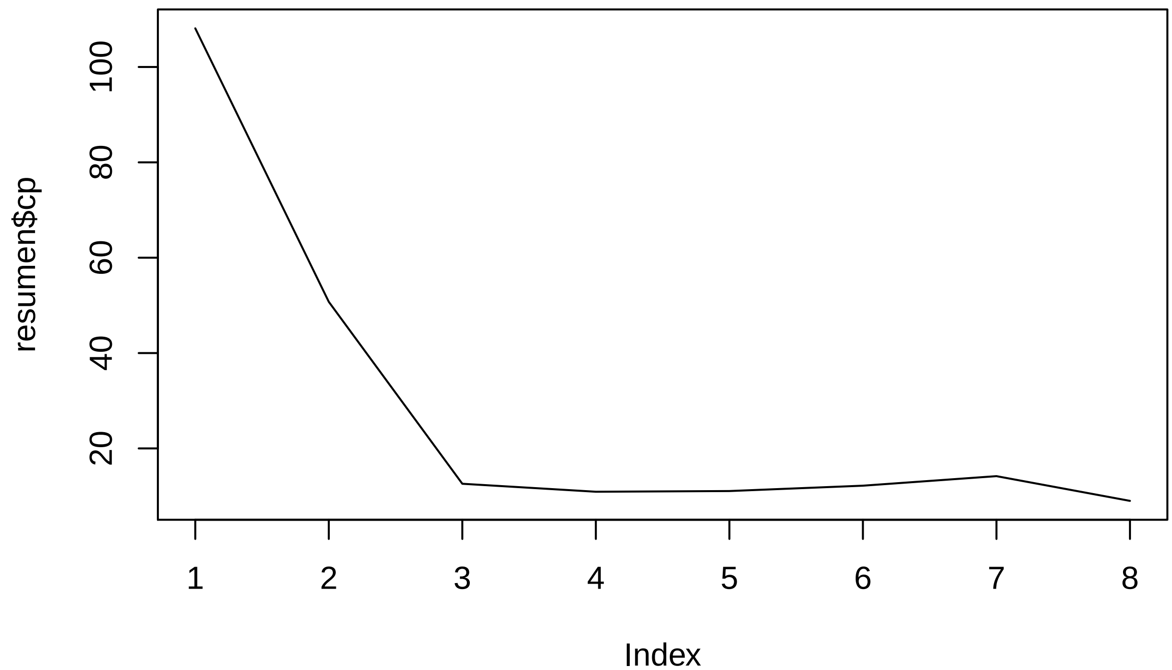
```
resumen$rsq
```

```
## [1] 0.2652052 0.3508716 0.4088450 0.4141466 0.4168209 0.4180876 0.4180884
## [8] 0.4284539
```

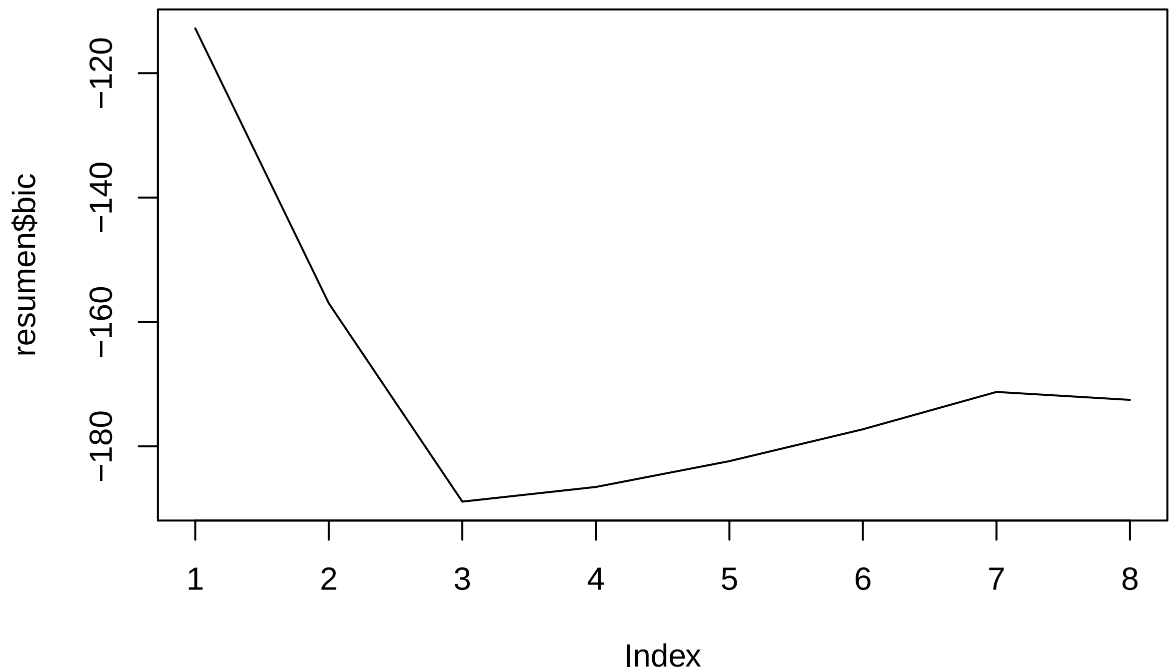
```
# Representación grafica
plot(resumen$adjr2, type="l")
```




```
plot(resumen$cp, type="l")
```




```r
plot(resumen$bic, type="l")
```




```r
which.min(resumen$cp)
```

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

```r
which.min(resumen$bic)
```

```
## [1] 3
```

```r
compos<- which.min(resumen$bic)

# Variables seleccionadas
vsel<- colnames(resumen$which)[resumen$which[compos,]]
vsel
```

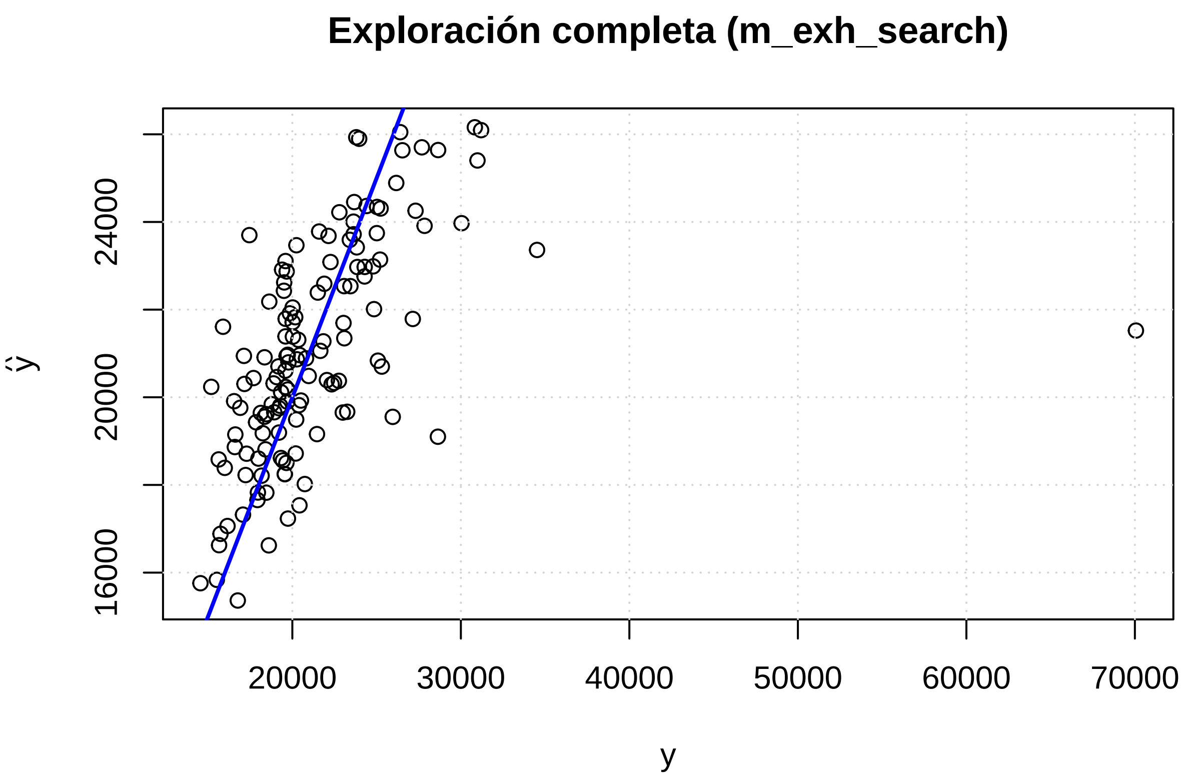
```
## [1] "(Intercept)" "managers"    "middleempl"  "employrate"
```

```r
# Se elimina el término independiente (Intercept)
vsel=vsel[-1]
formula <- as.formula(paste("income ~ ", paste(vsel, collapse= "+")))
formula
```

```
## income ~ managers + middleempl + employrate
```

```r
# Modelo resultante
m_exh_search<- lm(formula, data=employ_train)

# Cálculo de las predicciones
pred_exh_search=predict(m_exh_search, newdata=employ_test)
```

```
# Medida del ajuste
(ajuste_exh_search=Ajuste(employ_test$income, pred_exh_search, n_test, compos, "Exploración completa (m_
```




```
## $MSE
## [1] 24346029
##
## $RMSE
## [1] 4934.17
##
## $R2
## [1] 0.2355913
##
## $R2_ajust
## [1] 0.2180858
```

Nuevamente obtenemos un R2 ajustado bajo, el modelo no se ajusta bien a los datos.

2.3.3 Modelo de regresión lineal con selección de variables mediante algortimos genéticos

```
# La variable respuesta es el salario
xent <- as.matrix(employment[index_train, names(employment)!="income"])
yent <- employment[index_train, "income"]
```

```r
# Función de actitud para maximizar
fitness <- function(string)
{
  inc <- which(string==1)
  X <- cbind(1, xent[,inc])
  mod <- lm.fit(X, yent)
  class(mod) <- "lm"
  -AIC(mod)
}


# Modelo
AG <- ga("binary", fitness = fitness, nBits = ncol(xent), names = colnames(xent))

summary(AG)
```

```
## +-----------------------------------+
## |         Genetic Algorithm         |
## +-----------------------------------+
##
## GA settings:
## Type                  =  binary
## Population size       =  50
## Number of generations =  100
## Elitism               =  2
## Crossover probability =  0.8
## Mutation probability  =  0.1
##
## GA results:
## Iterations             = 100
## Fitness function value = -7687.381
## Solution =
##      farmers tradesmen managers workers unemployed middleempl retired
## [1,]       1         1        1       1          1          1       1
##      employrate
## [1,]          1
```

```r
# Ajuste del modelo resultante
posicvariables=which(AG@solution==1)
datos_sel=data.frame(income=employment[,"income"],
                     employment[,posicvariables])


summary(datos_sel)
```

```
##      income         farmers          tradesmen         managers
##  Min.   :12187   Min.   : 0.0000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:18367   1st Qu.: 0.5025   1st Qu.: 2.780   1st Qu.: 2.825
##  Median :19990   Median : 1.9550   Median : 4.000   Median : 4.650
##  Mean   :21003   Mean   : 3.3544   Mean   : 4.204   Mean   : 5.286
##  3rd Qu.:22768   3rd Qu.: 4.6125   3rd Qu.: 5.312   3rd Qu.: 7.143
##  Max.   :70062   Max.   :29.0300   Max.   :16.130   Max.   :22.730
##     workers        unemployed      middleempl        retired
##  Min.   : 7.69   Min.   : 0.00   Min.   : 0.000   Min.   : 9.33
##  1st Qu.:28.64   1st Qu.:11.23   1st Qu.: 8.547   1st Qu.:23.23
```

```
##  Median :33.67   Median :13.55   Median :11.905   Median :27.45
##  Mean   :33.65   Mean   :13.35   Mean   :12.005   Mean   :28.16
##  3rd Qu.:38.41   3rd Qu.:15.55   3rd Qu.:15.465   3rd Qu.:32.14
##  Max.   :57.14   Max.   :29.19   Max.   :31.580   Max.   :51.28
##    employrate
##  Min.   : 75.08
##  1st Qu.: 88.35
##  Median : 90.66
##  Mean   : 90.31
##  3rd Qu.: 92.70
##  Max.   :100.00
```

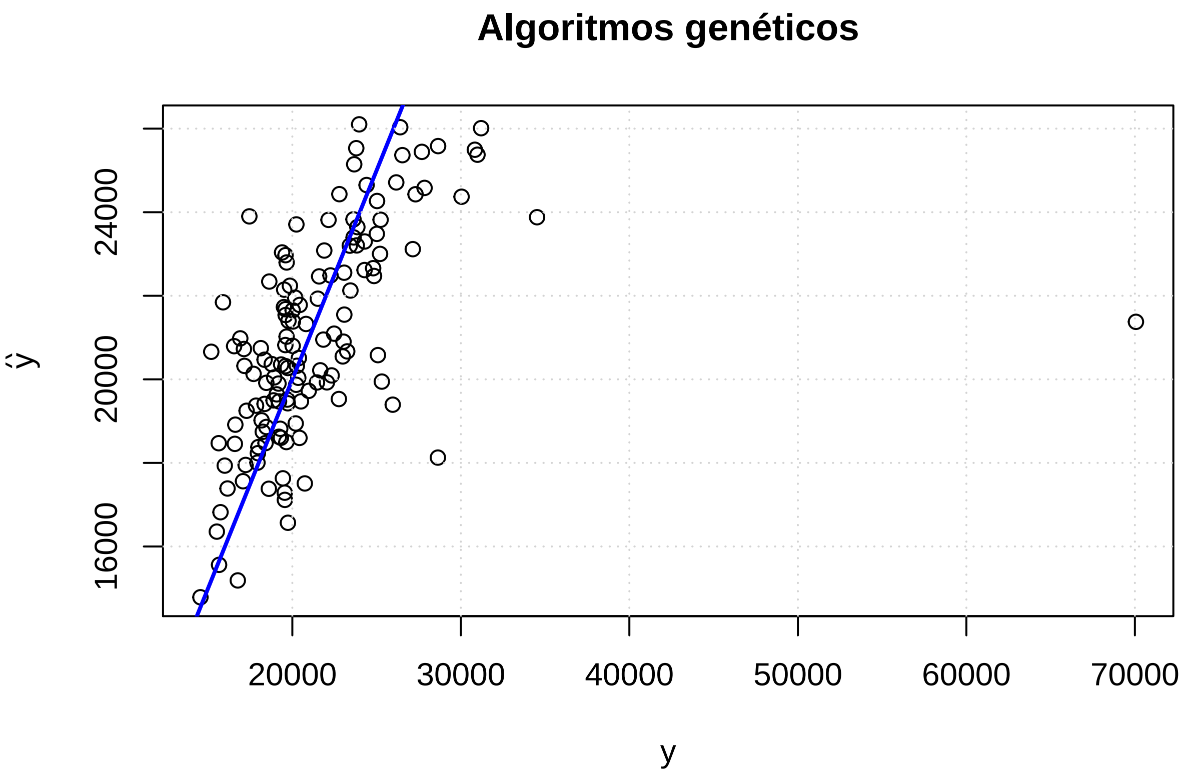
```r
modeloAG=lm(income~., data=datos_sel[index_train,])
summary(modeloAG)
```

```
##
## Call:
## lm(formula = income ~ ., data = datos_sel[index_train, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11434.3  -1679.6   -316.5   1501.5  15630.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.415e+06  2.016e+06  -2.686  0.00753 **
## farmers      5.408e+04  2.016e+04   2.683  0.00760 **
## tradesmen    5.416e+04  2.016e+04   2.687  0.00752 **
## managers     5.444e+04  2.016e+04   2.701  0.00722 **
## workers      5.403e+04  2.016e+04   2.680  0.00767 **
## unemployed   5.402e+04  2.016e+04   2.680  0.00767 **
## middleempl   5.428e+04  2.016e+04   2.692  0.00739 **
## retired      5.405e+04  2.016e+04   2.681  0.00765 **
## employrate   2.991e+02  5.407e+01   5.532 5.76e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3160 on 396 degrees of freedom
## Multiple R-squared:  0.4285, Adjusted R-squared:  0.4169
## F-statistic: 37.11 on 8 and 396 DF,  p-value: < 2.2e-16
```

```r
AG.pred=predict(modeloAG, datos_sel[-index_train,])

dim(employ_test)
```

```
## [1] 135   9
```

```r
# Medida del ajuste
(ajuste_AG=Ajuste(employ_test$income, AG.pred, n_test, ncol(datos_sel), "Algoritmos genéticos"))
```




```
## $MSE
## [1] 24630697
##
## $RMSE
## [1] 4962.932
##
## $R2
## [1] 0.227129
##
## $R2_ajust
## [1] 0.1714822
```

```
# La variable respuesta es el salario
xent <- as.matrix(employment[index_train, names(employment)!="income"])
yent <- employment[index_train, "income"]


# Función de actitud para maximizar
fitness <- function(string)
{
  inc <- which(string==1)
  X <- cbind(1, xent[,inc])
  mod <- lm.fit(X, yent)
  class(mod) <- "lm"
  -AIC(mod)
}
```

```
# Modelo
AG <- ga("binary", fitness = fitness, nBits = ncol(xent), names = colnames(xent))

summary(AG)
```

```
## +-----------------------------------+
## |         Genetic Algorithm         |
## +-----------------------------------+
##
## GA settings:
## Type                  =  binary
## Population size       =  50
## Number of generations =  100
## Elitism               =  2
## Crossover probability =  0.8
## Mutation probability  =  0.1
##
## GA results:
## Iterations             = 100
## Fitness function value = -7689.395
## Solution =
##      farmers tradesmen managers workers unemployed middleempl retired
## [1,]       0         1        1       0          0          1       0
##      employrate
## [1,]          1
```

```
# Ajuste del modelo resultante
posicvariables=which(AG@solution==1)
datos_sel=data.frame(income=employment[,"income"],
                     employment[,posicvariables])


summary(datos_sel)
```

```
##      income        tradesmen         managers        middleempl
##  Min.   :12187   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:18367   1st Qu.: 2.780   1st Qu.: 2.825   1st Qu.: 8.547
##  Median :19990   Median : 4.000   Median : 4.650   Median :11.905
##  Mean   :21003   Mean   : 4.204   Mean   : 5.286   Mean   :12.005
##  3rd Qu.:22768   3rd Qu.: 5.312   3rd Qu.: 7.143   3rd Qu.:15.465
##  Max.   :70062   Max.   :16.130   Max.   :22.730   Max.   :31.580
##    employrate
##  Min.   : 75.08
##  1st Qu.: 88.35
##  Median : 90.66
##  Mean   : 90.31
##  3rd Qu.: 92.70
##  Max.   :100.00
```

```
modeloAG=lm(income~., data=datos_sel[index_train,])
summary(modeloAG)
```

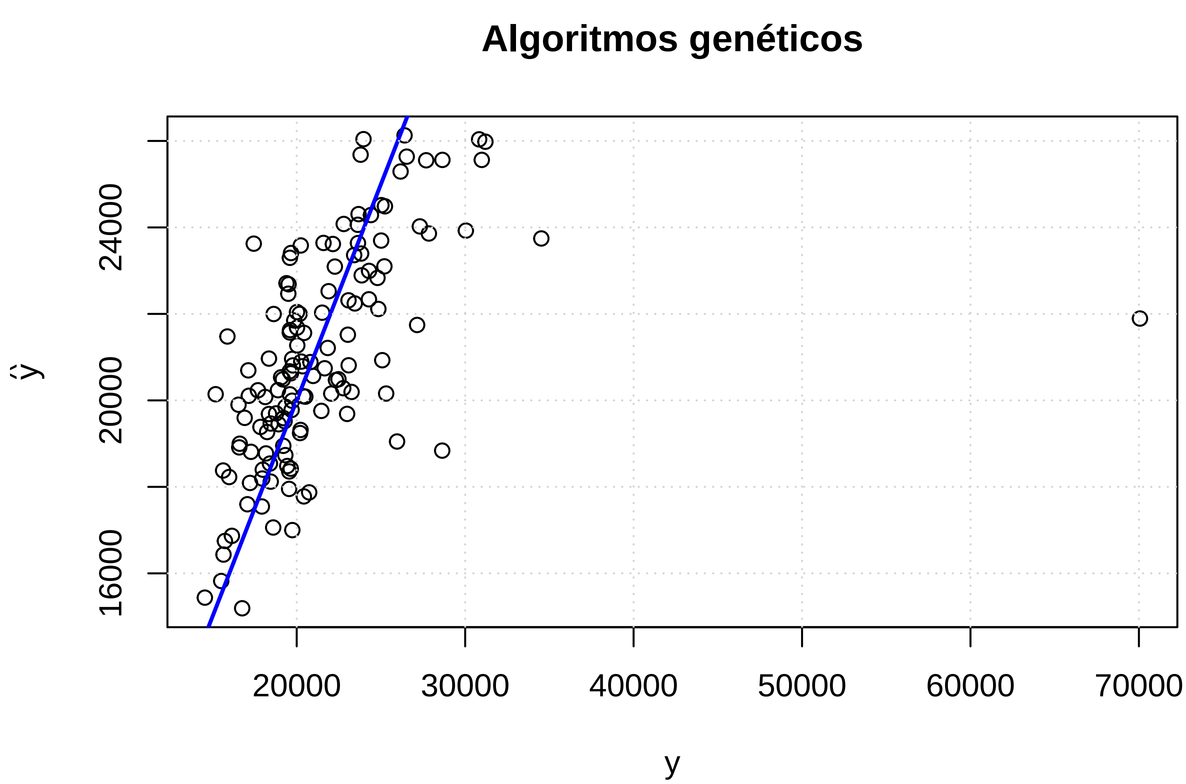
```
##
## Call:
## lm(formula = income ~ ., data = datos_sel[index_train, ])
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11605.9  -1789.8   -401.1   1445.6  17046.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12707.17    4432.06  -2.867  0.00436 **
## tradesmen      121.09      63.65   1.903  0.05782 .
## managers       402.25      48.16   8.352 1.12e-15 ***
## middleempl     225.94      33.36   6.772 4.55e-11 ***
## employrate     312.75      50.40   6.205 1.37e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3183 on 400 degrees of freedom
## Multiple R-squared:  0.4141, Adjusted R-squared:  0.4083
## F-statistic: 70.69 on 4 and 400 DF,  p-value: < 2.2e-16
```

```
AG.pred=predict(modeloAG, datos_sel[-index_train,])

dim(employ_test)
```

```
## [1] 135   9
```

```
# Medida del ajuste
(ajuste_AG=Ajuste(employ_test$income, AG.pred, n_test, ncol(datos_sel), "Algoritmos genéticos"))
```



```
## $MSE
## [1] 24066933
##
## $RMSE
## [1] 4905.806
##
## $R2
## [1] 0.2465055
##
## $R2_ajust
## [1] 0.2173003
```

La primera vez que aplico algoritmos genéticos no consigo reducir variables ni mejorar el R2, pero aplicándolo 2 veces sí selecciona variables.

2.3.4 Resultados y conclusiones

Construimos una tabla resumen de todos los procedimientos de selección de variables utilizados en este ejercicio para poder comparar los resultados obtenidos y sacar conclusiones.

```r
table_full=c(ajuste_full$MSE, ajuste_full$RMSE, ajuste_full$R2, ajuste_full$R2_ajust)

table_AG=c(ajuste_AG$MSE, ajuste_AG$RMSE, ajuste_AG$R2, ajuste_AG$R2_ajust )

table_exh=c(ajuste_exh_search$MSE, ajuste_exh_search$RMSE,
            ajuste_exh_search$R2, ajuste_exh_search$R2_ajust)


tabla_resumen = data.frame (round(rbind(table_full, table_AG, table_exh), 3),
                            row.names=c("Modelo completo",
                                        "Modelo con algoritmos genéticos",
                                        "Modelo búsqueda exhaustiva"))

print(knitr::kable(tabla_resumen, format = "pandoc",
                   col.names = c("MSE", "RMSE", "R2", "R2_ajust"), align='c'))
```

```
##
##
##                                       MSE         RMSE       R2      R2_ajust
## --------------------------------  ----------  ----------  -------  ----------
## Modelo completo                    24630697    4962.932    0.227     0.178
## Modelo con algoritmos genéticos    24066933    4905.806    0.247     0.217
## Modelo búsqueda exhaustiva         24346029    4934.170    0.236     0.218
```

Obtenemos un error alto y un R2 ajustado bajo para los 3 procedimientos de selección de variables. De entre los 3 utilizados en este ejecicio el que mejores resultados ofrece es el de búsqueda exhaustiva.