Laboratorio 6, Tópicos en análisis datos 1

Joshua Isaac Cervantes Artavia

2023-10-14

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
library(tidyverse) # To manipulate data, and other things
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.2     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.2     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggforce) # To use ggcircle
library(openxlsx) # To manipulate Excel
library(kableExtra) # To make beatiful tables
```

```
Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':

    group_rows
```

```
library(ade4)
```

```r
tryCatch(
    {
        # Directorio donde se ubica el qmd
        directory <- dirname(rstudioapi::getSourceEditorContext()$path)
        setwd(directory) # Establecer el directorio del archivo como la raiz
    },
    error = function(e) {
        message("")
        print("")
    }
)
```

```
[1] ""
```

1 Análisis de componentes principales

```r
data(USArrests)

# We make the pca
acp_res <- dudi.pca(USArrests, scannf = FALSE, nf = 3)


# We check that type of the object acp_res
is.dudi(acp_res)
```

```
[1] TRUE
```

```r
# We check the rank of the matrix
acp_res$rank
```

```
[1] 4
```

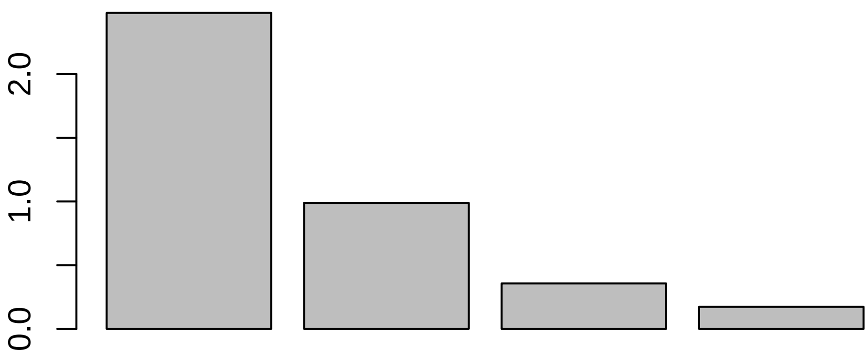
```r
# We check the number of factors
acp_res$nf
```

```
[1] 3
```

```r
# We check the eigen values
acp_res$eig
```

```
[1] 2.4802416 0.9897652 0.3565632 0.1734301
```

```r
# We do a bar plot
barplot(acp_res$eig)
```




```r
# We check the table centered, and also standarized
acp_res$tab
```

```
                Murder    Assault    UrbanPop         Rape
Alabama     1.25517927 0.79078716 -0.52619514 -0.003451159
Alaska      0.51301858 1.11805959 -1.22406668  2.509423922
Arizona     0.07236067 1.49381682  1.00912225  1.053466257
Arkansas    0.23470832 0.23321191 -1.08449238 -0.186793976
California  0.28109336 1.27563520  1.77678094  2.088813930
```

Colorado	0.02597562	0.40290872	0.86954794	1.883901370
Connecticut	-1.04088037	-0.73648418	0.79976079	-1.092723190
Delaware	-0.43787481	0.81502956	0.45082502	-0.585834225
Florida	1.76541475	1.99078607	1.00912225	1.150530102
Georgia	2.22926518	0.48775713	-0.38662083	0.492652934
Hawaii	-0.57702994	-1.51224105	1.21848371	-0.111299875
Idaho	-1.20322802	-0.61527217	-0.80534376	-0.758392170
Illinois	0.60578867	0.94836277	1.21848371	0.298525246
Indiana	-0.13637203	-0.70012057	-0.03768506	-0.025020902
Iowa	-1.29599811	-1.39102904	-0.59598230	-1.071153447
Kansas	-0.41468229	-0.67587817	0.03210209	-0.348567050
Kentucky	0.44344101	-0.74860538	-0.94491807	-0.531909867
Louisiana	1.76541475	0.94836277	0.03210209	0.104397557
Maine	-1.31919063	-1.06375661	-1.01470522	-1.448623952
Maryland	0.81452136	1.56654403	0.10188925	0.708350366
Massachusetts	-0.78576263	-0.26375734	1.35805802	-0.531909867
Michigan	1.00006153	1.02108998	0.59039932	1.495645992
Minnesota	-1.18003550	-1.19708982	0.03210209	-0.682898069
Mississippi	1.92776240	1.06957478	-1.50321530	-0.445630894
Missouri	0.28109336	0.08775750	0.31125071	0.751489853
Montana	-0.41468229	-0.74860538	-0.87513091	-0.521124995
Nebraska	-0.80895515	-0.83345379	-0.24704653	-0.510340124
Nevada	1.02325405	0.98472638	1.07890940	2.671196996
New Hampshire	-1.31919063	-1.37890783	-0.66576945	-1.265281135
New Jersey	-0.08998698	-0.14254532	1.63720664	-0.262288077
New Mexico	0.83771388	1.38472601	0.31125071	1.172099845
New York	0.76813632	1.00896878	1.42784517	0.525007549
North Carolina	1.20879423	2.01502847	-1.43342815	-0.553479610
North Dakota	-1.62069341	-1.52436225	-1.50321530	-1.502548310
Ohio	-0.11317951	-0.61527217	0.66018648	0.018118584
Oklahoma	-0.27552716	-0.23951493	0.17167640	-0.132869618
Oregon	-0.66980002	-0.14254532	0.10188925	0.870123440
Pennsylvania	-0.34510472	-0.78496898	0.45082502	-0.682898069
Rhode Island	-1.01768785	0.03927269	1.49763233	-1.394699594
South Carolina	1.53348953	1.31199880	-1.22406668	0.136752172
South Dakota	-0.92491776	-1.02739300	-1.43342815	-0.909380373
Tennessee	1.25517927	0.20896951	-0.45640799	0.611286522
Texas	1.13921666	0.36654512	1.00912225	0.460298320
Utah	-1.06407289	-0.61527217	1.00912225	0.179891658
Vermont	-1.29599811	-1.48799864	-2.34066115	-1.081938318
Virginia	0.16513075	-0.17890893	-0.17725937	-0.057375517
Washington	-0.87853272	-0.31224214	0.52061217	0.535792421
West Virginia	-0.48425985	-1.08799901	-1.85215107	-1.286850878

```
Wisconsin      -1.20322802 -1.42739264  0.03210209 -1.125077805
Wyoming        -0.22914211 -0.11830292 -0.38662083 -0.607403968
```

```
# We check the weights of the columns
acp_res$cw
```

```
[1] 1 1 1 1
```

```
# We get the weight of the rows
acp_res$lw
```

```
 [1] 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02
[16] 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02
[31] 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02
[46] 0.02 0.02 0.02 0.02 0.02
```

```
# We get the coordinates of the rows
acp_res$li
```

```
                     Axis1       Axis2       Axis3
Alabama        -0.98556588  1.13339238 -0.44426879
Alaska         -1.95013775  1.07321326  2.04000333
Arizona        -1.76316354 -0.74595678  0.05478082
Arkansas        0.14142029  1.11979678  0.11457369
California     -2.52398013 -1.54293399  0.59855680
Colorado       -1.51456286 -0.98755509  1.09500699
Connecticut     1.35864746 -1.08892789 -0.64325757
Delaware       -0.04770931 -0.32535892 -0.71863294
Florida        -3.01304227  0.03922851 -0.57682949
Georgia        -1.63928304  1.27894240 -0.34246008
Hawaii          0.91265715 -1.57046001  0.05078189
Idaho           1.63979985  0.21097292  0.25980134
Illinois       -1.37891072 -0.68184119 -0.67749564
Indiana         0.50546136 -0.15156254  0.22805484
Iowa            2.25364607 -0.10405407  0.16456432
Kansas          0.79688112 -0.27016470  0.02555331
Kentucky        0.75085907  0.95844029 -0.02836942
Louisiana      -1.56481798  0.87105466 -0.78348036
```

```
Maine           2.39682949  0.37639158 -0.06568239
Maryland       -1.76336939  0.42765519 -0.15725013
Massachusetts   0.48616629 -1.47449650 -0.60949748
Michigan       -2.10844115 -0.15539682  0.38486858
Minnesota       1.69268181 -0.63226125  0.15307043
Mississippi    -0.99649446  2.39379599 -0.74080840
Missouri       -0.69678733 -0.26335479  0.37744383
Montana         1.18545191  0.53687437  0.24688932
Nebraska        1.26563654 -0.19395373  0.17557391
Nevada         -2.87439454 -0.77560020  1.16338049
New Hampshire   2.38391541 -0.01808229  0.03685539
New Jersey     -0.18156611 -1.44950571 -0.76445355
New Mexico     -1.98002375  0.14284878  0.18369218
New York       -1.68257738 -0.82318414 -0.64307509
North Carolina -1.12337861  2.22800338 -0.86357179
North Dakota    2.99222562  0.59911882  0.30127728
Ohio            0.22596542 -0.74223824 -0.03113912
Oklahoma        0.31178286 -0.28785421 -0.01530979
Oregon         -0.05912208 -0.54141145  0.93983298
Pennsylvania    0.88841582 -0.57110035 -0.40062871
Rhode Island    0.86377206 -1.49197842 -1.36994570
South Carolina -1.32072380  1.93340466 -0.30053779
South Dakota    1.98777484  0.82334324  0.38929333
Tennessee      -0.99974168  0.86025130  0.18808295
Texas          -1.35513821 -0.41248082 -0.49206886
Utah            0.55056526 -1.47150461  0.29372804
Vermont         2.80141174  1.40228806  0.84126309
Virginia        0.09633491  0.19973529  0.01171254
Washington      0.21690338 -0.97012418  0.62487094
West Virginia   2.10858541  1.42484670  0.10477467
Wisconsin       2.07971417 -0.61126862 -0.13886500
Wyoming         0.62942666  0.32101297 -0.24065923
```

```r
# We get the coordinates standarized of the rows
acp_res$l1
```

```
                       RS1         RS2         RS3
Alabama        -0.62580448  1.13923733 -0.74400791
Alaska         -1.23827840  1.07874786  3.41635214
Arizona        -1.11955544 -0.74980371  0.09174033
Arkansas        0.08979760  1.12557163  0.19187423
```

California	-1.60265093	-1.55089097	1.00239091
Colorado	-0.96170154	-0.99264795	1.83378597
Connecticut	0.86269998	-1.09454354	-1.07725038
Delaware	-0.03029397	-0.32703681	-1.20347999
Florida	-1.91319057	0.03943082	-0.96600463
Georgia	-1.04089508	1.28553796	-0.57351094
Hawaii	0.57950964	-1.57855894	0.08504341
Idaho	1.04122324	0.21206091	0.43508403
Illinois	-0.87556654	-0.68535747	-1.13458819
Indiana	0.32095265	-0.15234416	0.38191881
Iowa	1.43099699	-0.10459068	0.27559252
Kansas	0.50599537	-0.27155796	0.04279360
Kentucky	0.47677277	0.96338300	-0.04750969
Louisiana	-0.99361202	0.87554672	-1.31207864
Maine	1.52191412	0.37833264	-0.10999696
Maryland	-1.11968615	0.42986063	-0.26334359
Massachusetts	0.30870087	-1.48210054	-1.02071304
Michigan	-1.33879626	-0.15619821	0.64453160
Minnesota	1.07480168	-0.63552185	0.25634395
Mississippi	-0.63274380	2.40614090	-1.24061677
Missouri	-0.44243885	-0.26471292	0.63209751
Montana	0.75272605	0.53964306	0.41346053
Nebraska	0.80364087	-0.19495396	0.29403005
Nevada	-1.82515346	-0.77960000	1.94828968
New Hampshire	1.51371407	-0.01817554	0.06172098
New Jersey	-0.11528898	-1.45698088	-1.28021483
New Mexico	-1.25725511	0.14358546	0.30762556
New York	-1.06838567	-0.82742934	-1.07694480
North Carolina	-0.71331139	2.23949329	-1.44620613
North Dakota	1.89997263	0.60220851	0.50454295
Ohio	0.14348120	-0.74606599	-0.05214805
Oklahoma	0.19797267	-0.28933868	-0.02563900
Oregon	-0.03754073	-0.54420354	1.57391920
Pennsylvania	0.56411714	-0.57404554	-0.67092476
Rhode Island	0.54846910	-1.49967262	-2.29422023
South Carolina	-0.83861961	1.94337532	-0.50330453
South Dakota	1.26217681	0.82758926	0.65194163
Tennessee	-0.63480569	0.86468766	0.31497869
Texas	-0.86047171	-0.41460800	-0.82405772
Utah	0.34959226	-1.47909322	0.49190038
Vermont	1.77881160	1.40951972	1.40884621
Virginia	0.06116975	0.20076533	0.01961476
Washington	0.13772708	-0.97512716	1.04645867

```
West Virginia   1.33888786  1.43219470  0.17546401
Wisconsin       1.32055550 -0.61442096 -0.23255439
Wyoming         0.39966687  0.32266845 -0.40302713
```

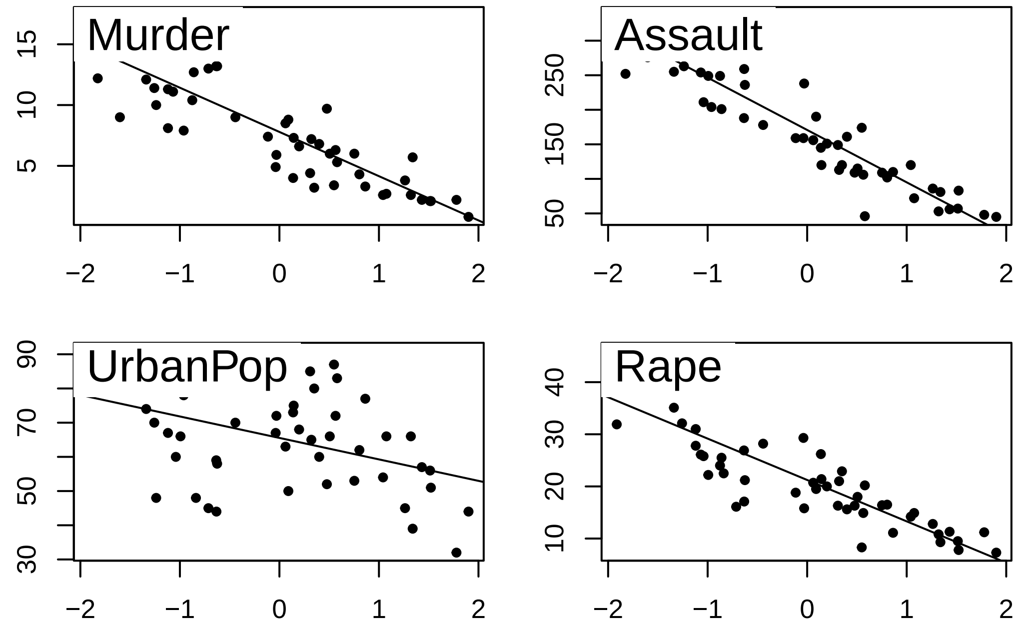
```r
# We get the coordinates of the columns
acp_res$co
```

```
              Comp1      Comp2      Comp3
Murder   -0.8439764  0.4160354 -0.2037600
Assault  -0.9184432  0.1870211 -0.1601192
UrbanPop -0.4381168 -0.8683282 -0.2257242
Rape     -0.8558394 -0.1664602  0.4883190
```

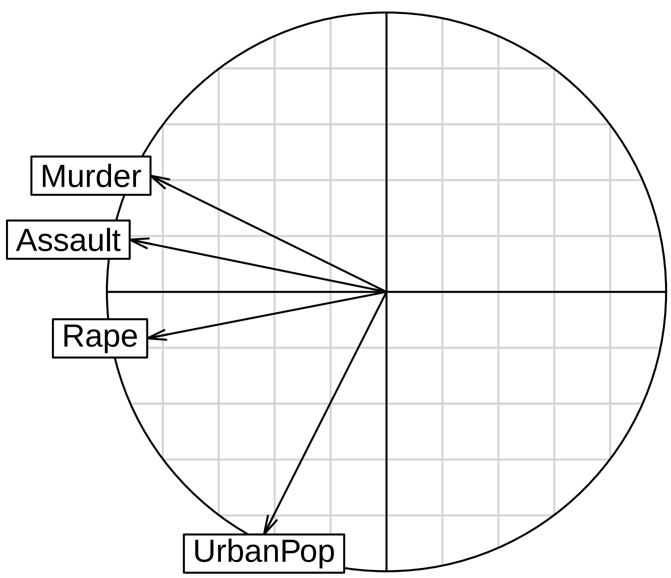
```r
# We get the coordinates standarized of the columns
acp_res$c1
```

```
                CS1        CS2        CS3
Murder   -0.5358995  0.4181809 -0.3412327
Assault  -0.5831836  0.1879856 -0.2681484
UrbanPop -0.2781909 -0.8728062 -0.3780158
Rape     -0.5434321 -0.1673186  0.8177779
```

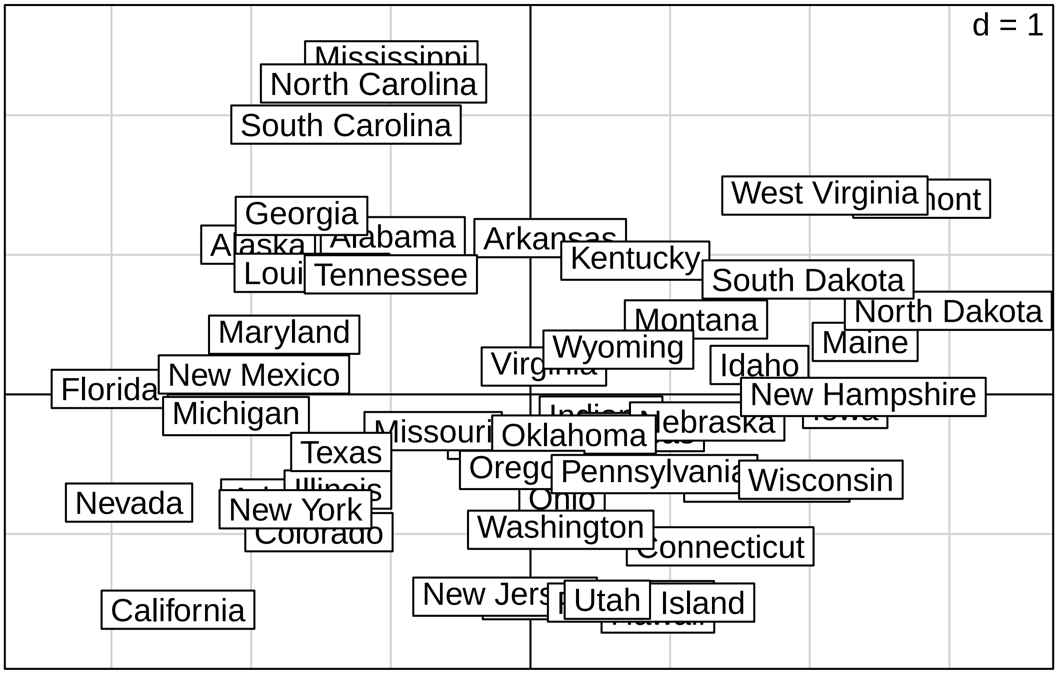
```r
# We get a plot of the relation between the coordinates of the pca 1 vs variables
score(acp_res)
```



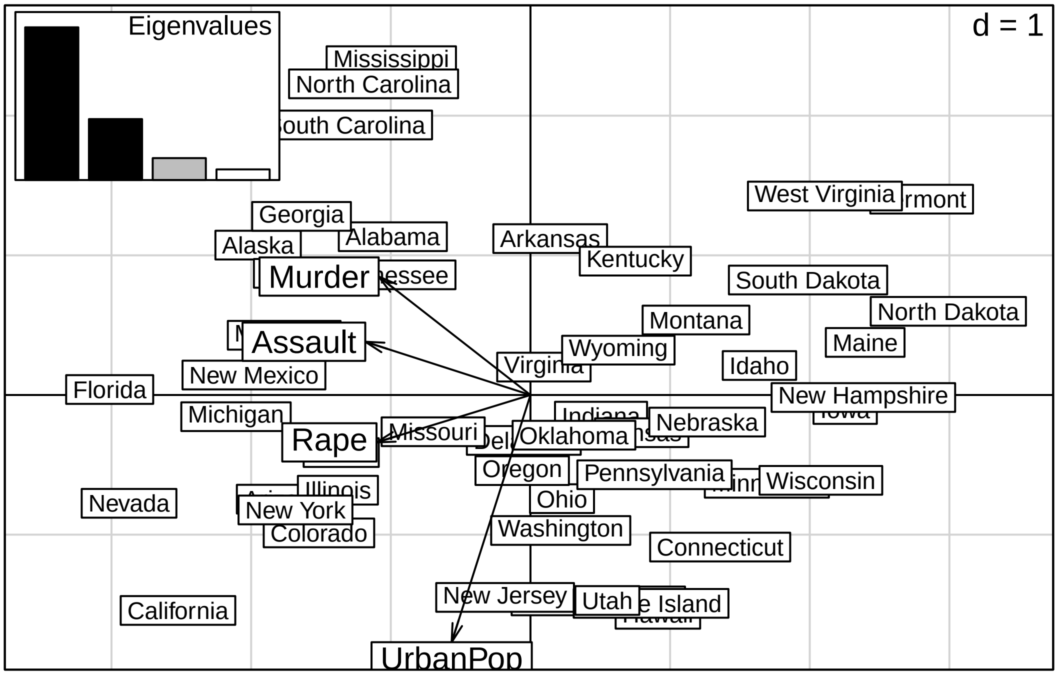

```r
# Correlations circle
s.corcircle(acp_res$co)
```



```r
# Principal plane
s.label(acp_res$li)
```




```r
# Simmultanious correlation variables, and principal plane and value of eigen values
scatter(acp_res)
```



2 Análisis de Correspondencias Simples

```r
# We load the data
data(housetasks)
```

```r
# We make a chi square test
chisq.test(housetasks)
```

```
	Pearson's Chi-squared test

data:  housetasks
X-squared = 1944.5, df = 36, p-value < 2.2e-16
```

Se recha la hipótesis nula de que no existe relación entre las variables.

```r
# We estimate the row profiles
round(housetasks / apply(housetasks, 1, sum), 2)
```

```
           Wife Alternating Husband Jointly
```

```
Laundry    0.89        0.08    0.01    0.02
Main_meal  0.81        0.13    0.03    0.03
Dinner     0.71        0.10    0.06    0.12
Breakfeast 0.59        0.26    0.11    0.05
Tidying    0.43        0.09    0.01    0.47
Dishes     0.28        0.21    0.04    0.47
Shopping   0.28        0.19    0.07    0.46
Official   0.12        0.48    0.24    0.16
Driving    0.07        0.37    0.54    0.02
Finances   0.12        0.12    0.19    0.58
Insurance  0.06        0.01    0.38    0.55
Repairs    0.00        0.02    0.97    0.01
Holidays   0.00        0.01    0.04    0.96
```

```r
# We estimate the column profiles
round(t(housetasks) / apply(t(housetasks), 1, sum), 2)
```

```
            Laundry Main_meal Dinner Breakfeast Tidying Dishes Shopping
Wife           0.26      0.21   0.13       0.14    0.09   0.05     0.06
Alternating    0.06      0.08   0.04       0.14    0.04   0.09     0.09
Husband        0.01      0.01   0.02       0.04    0.00   0.01     0.02
Jointly        0.01      0.01   0.03       0.01    0.11   0.10     0.11
            Official Driving Finances Insurance Repairs Holidays
Wife            0.02    0.02     0.02      0.01    0.00     0.00
Alternating     0.18    0.20     0.05      0.00    0.01     0.00
Husband         0.06    0.20     0.06      0.14    0.42     0.02
Jointly         0.03    0.01     0.13      0.15    0.00     0.30
```

```r
# We estimate the weight of the rows
round(apply(housetasks, 1, sum) / sum(housetasks), 2)
```

```
   Laundry  Main_meal     Dinner Breakfeast    Tidying     Dishes   Shopping
      0.10       0.09       0.06       0.08       0.07       0.06       0.07
  Official    Driving   Finances  Insurance    Repairs   Holidays
      0.06       0.08       0.06       0.08       0.09       0.09
```

```r
# We estima the weight of the columns
round(apply(housetasks, 2, sum) / sum(housetasks), 2)
```

```
      Wife Alternating      Husband      Jointly 
      0.34          0.15         0.22         0.29 
```

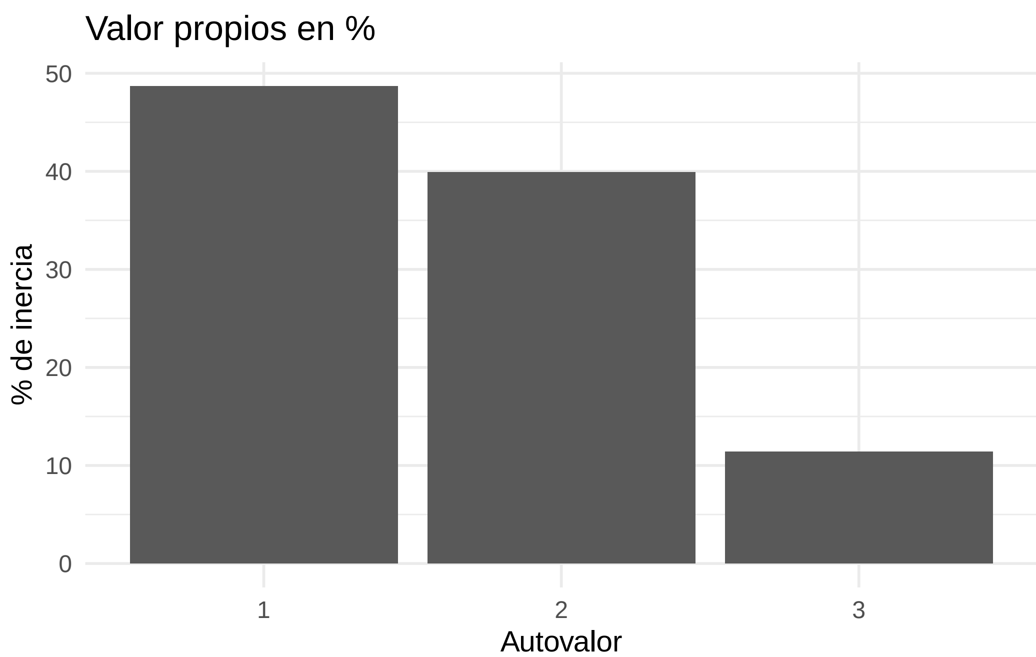
```
# We make a afc
z <- dudi.coa(df = housetasks, scannf = F, nf = 3)

# We get the eigen values
round(z$eig, 2)
```

```
[1] 0.54 0.45 0.13
```

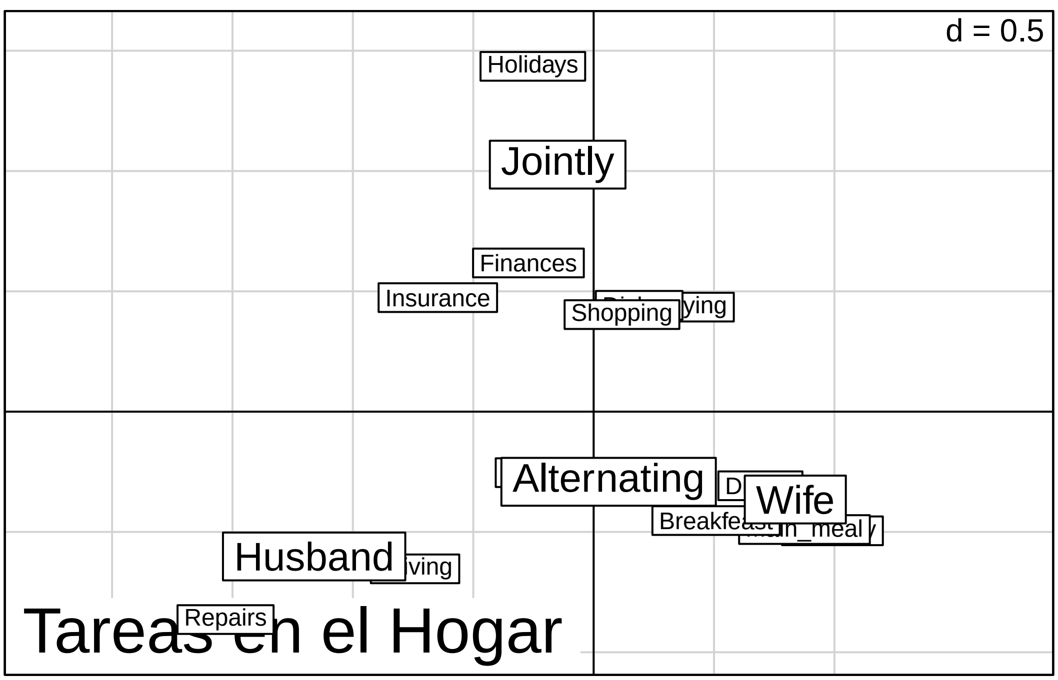
```
# Inertia explained by each eigen value
round(z$eig / sum(z$eig) * 100, 2)
```

```
[1] 48.69 39.91 11.40
```

```
# We make the plot of the eigen values to check the inertia, and the inflection point
inercia <- z$eig / sum(z$eig) * 100


# Bar plot
ggplot(, aes(x = c("1", "2", "3"), y = inercia)) +
    geom_bar(stat = "identity") +
    labs(x = "Autovalor", y = "% de inercia") +
    theme_minimal() +
    ggtitle("Valor propios en %")
```

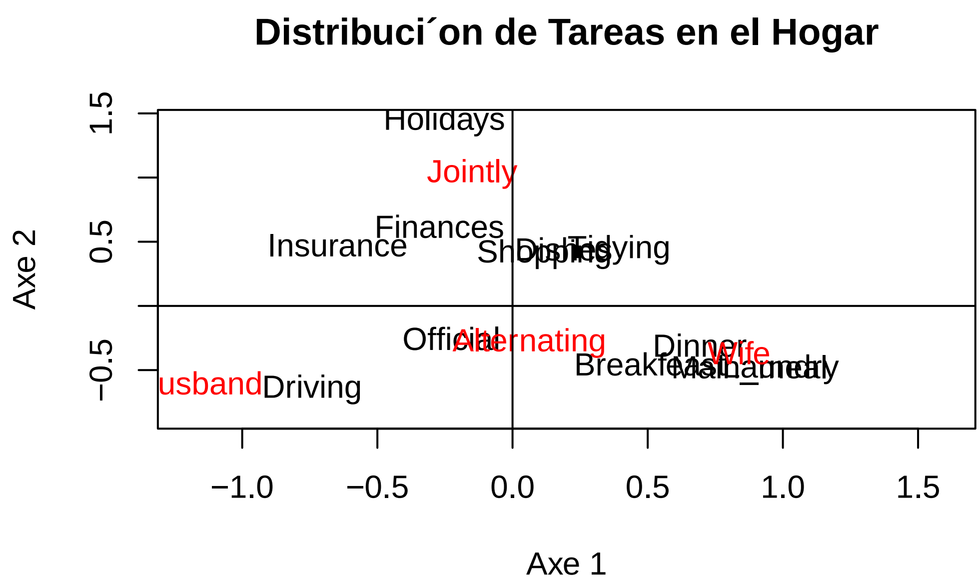



```
# Factorial plane
scatter(z, method = 1, sub = "Tareas en el Hogar", posieig = "none")
```



NULL

```r
# Second way to make factorial plane
plot(z$li[, 1], z$li[, 2], type = "n", xlab = "Axe 1", ylab = "Axe 2", xlim = c(-1.2, 1.6))
text(z$li[, 1], z$li[, 2], label = row.names(housetasks))
text(z$co[, 1], z$co[, 2], label = colnames(housetasks), col = "red")
title("Distribuci´on de Tareas en el Hogar")
abline(h = 0, v = 0)
```




```r
# Coordinates of rows profiles
round(z$li, 2)
```

```
           Axis1 Axis2 Axis3
Laundry     0.99 -0.50 -0.32
Main_meal   0.88 -0.49 -0.16
Dinner      0.69 -0.31 -0.21
Breakfeast  0.51 -0.45  0.22
Tidying     0.39  0.43 -0.09
Dishes      0.19  0.44  0.27
Shopping    0.12  0.40  0.20
Official   -0.23 -0.25  0.92
```

```
Driving    -0.74 -0.65  0.54
Finances   -0.27  0.62  0.03
Insurance  -0.65  0.47 -0.29
Repairs    -1.53 -0.86 -0.47
Holidays   -0.25  1.44 -0.13
```

```r
# Coordinates of columns profiles
round(z$co, 2)
```

```
            Comp1 Comp2 Comp3
Wife         0.84 -0.37 -0.20
Alternating  0.06 -0.29  0.85
Husband     -1.16 -0.60 -0.19
Jointly     -0.15  1.03 -0.05
```

```r
# We estimate the contribution of the columns to the inertia
inertia.dudi(z, col.inertia = T)$col.abs
```

```
                Axis1     Axis2      Axis3
Wife        44.462018 10.312237 10.8220753
Alternating  0.103739  2.782794 82.5492464
Husband     54.233879 17.786612  6.1331792
Jointly      1.200364 69.118357  0.4954991
```

```r
# We estimate the contribution of the rows to the inertia
inertia.dudi(z, row.inertia = T)$row.abs
```

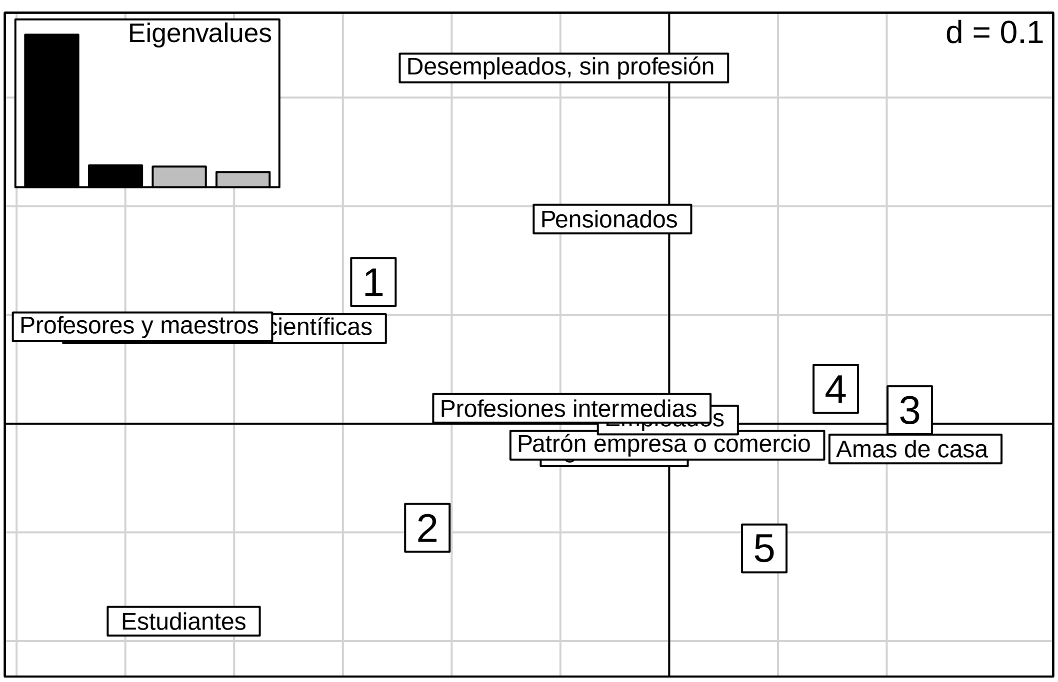
```
                Axis1      Axis2       Axis3
Laundry    18.2867003  5.5638913  7.96842443
Main_meal  12.3888433  4.7355230  1.85868941
Dinner      5.4713982  1.3210221  2.09692603
Breakfeast  3.8249284  3.6986131  3.06939857
Tidying     1.9983518  2.9656441  0.48873403
Dishes      0.4261663  2.8441170  3.63429434
Shopping    0.1755248  2.5151584  2.22335679
Official    0.5207837  0.7956201 36.94038942
Driving     8.0778371  7.6468564 18.59638635
Finances    0.8750075  5.5585460  0.06175066
```

```
Insurance    6.1470616  4.0203590  5.25263863
Repairs     40.7300940 15.8806509 16.59639139
Holidays     1.0773030 42.4539986  1.21261994
```

k)

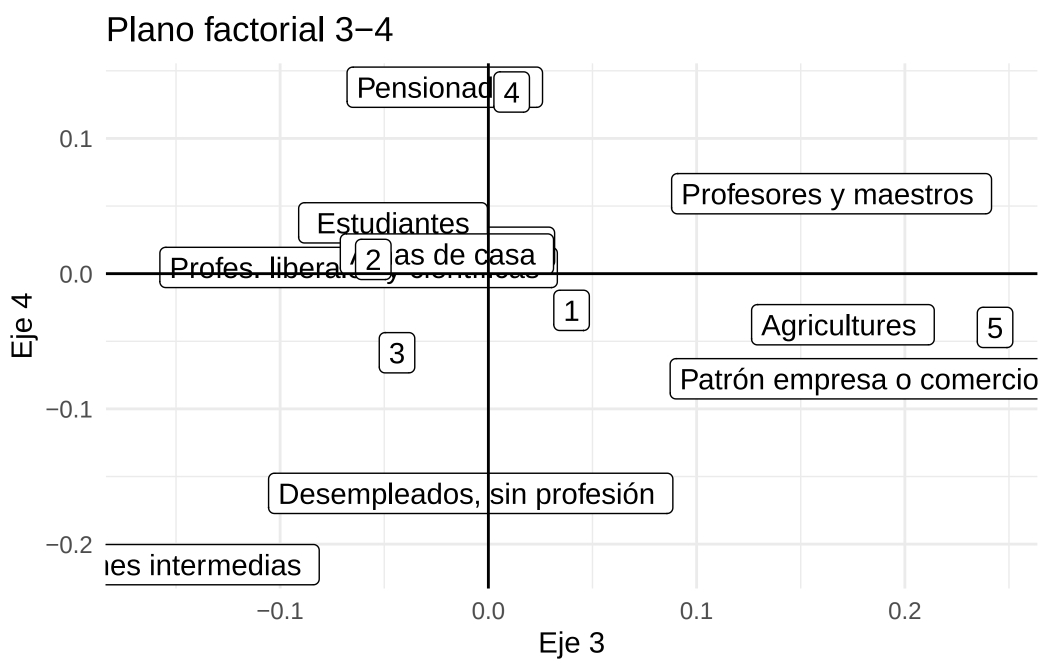
2.1 Ejercicio 3 capítulo 4

```r
# We read the data
profesiones <- read.xlsx("Ejercicios-Cap4.xlsx", 1)


# We make the afc

rownames(profesiones) <- profesiones[, 1]
profesiones <- profesiones[, -1]


profesiones_coa <- dudi.coa(df = profesiones, scannf = F, nf = 4)

scatter(profesiones_coa)
```



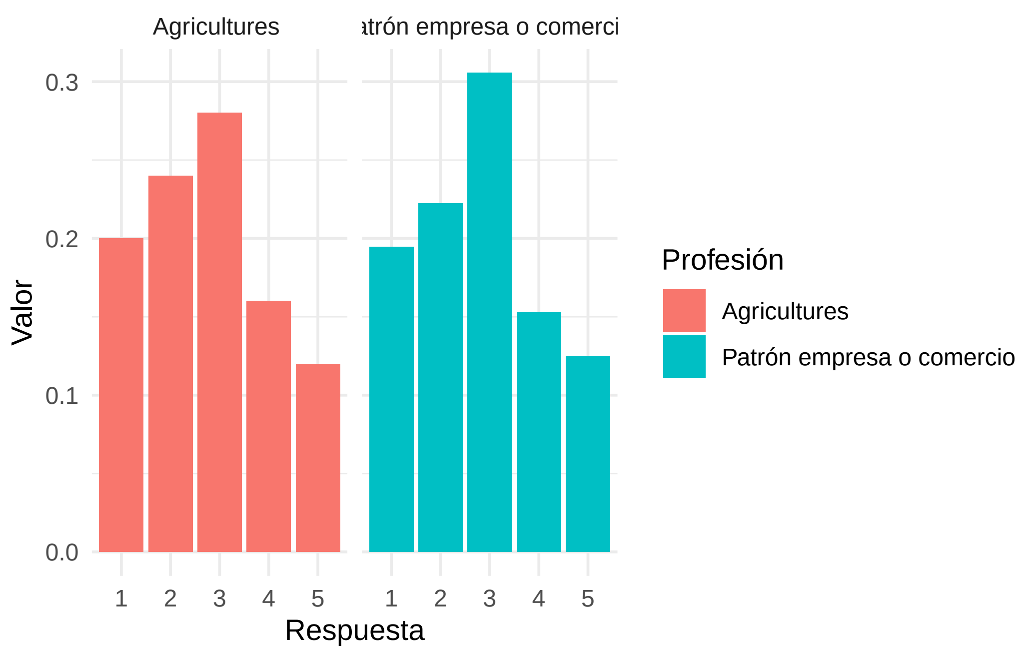
Se puede observar que las amas de casa claramente se oponene a los profesores y maestros, así como los profes. liberalres y científicas teniéndose que los maestros aportan en su gran mayoría a la respuesta de que está en total desacuerdo de con el "FMI ayude a resolver la crisis". Los otros que se oponen a estos últimos respecto al segundo eje y estos aportan en su gran mayoría con estar no tan de acuerdo. Además, en su gran mayoría las amas de casa aportan a la respuesta de podría estar de acuerdo. Las profesiones intermedias, empleados, patrón empresa o comercio, y agricultores no se ven tan bien representados en este plano.

```
# Factorial plane 3-4
ggplot(, ) +
    geom_label(aes(x = profesiones_coa$li[, 3], y = profesiones_coa$li[, 4], label = rownames(p
    geom_label(aes(x = profesiones_coa$co[, 3], y = profesiones_coa$co[, 4], label = colnames(p
    geom_vline(xintercept = 0) +
    geom_hline(yintercept = 0) +
    theme_minimal() +
    labs(x = "Eje 3", y = "Eje 4") +
    ggtitle("Plano factorial 3-4")
```



Se puede observar como los agricultores y los patrones de empresa o comercio se encuentran del mismo lado en el plano factorial 3-4. Vamos a graficar los perfiles fila mediante un barplot para ver si esto es cierto

```
# We estimate the row profiles
row_profiles_profesiones <- profesiones / apply(profesiones, 1, sum)
row_profiles_profesiones$profesion <- rownames(row_profiles_profesiones)
row_profiles_profesiones <- row_profiles_profesiones %>% gather(key = "Respuesta", value = "Val

row_profiles_profesiones %>%
    filter(profesion %in% c("Agricultures ", "Patrón empresa o comercio ")) %>%
    ggplot(aes(x = Respuesta, y = Valor, fill = profesion)) +
    geom_bar(stat = "identity") +
    facet_wrap(. ~ profesion) +
    theme_minimal() +
    labs(fill = "Profesión", x = "Respuesta", y = "Valor")
```



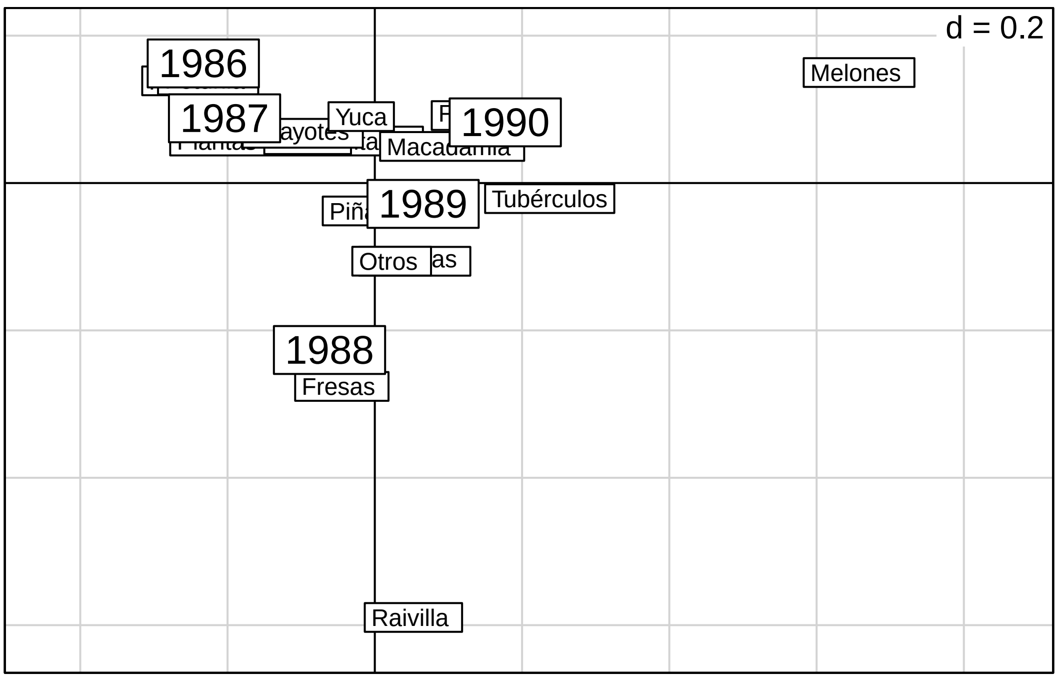
Se puede observar que de cierta forma sí tienen un comportamiento similar tanto los agricultores omo los patrones de empresas o comercios. Tal y como lo proponía el plano factorial 3-4.

2.2 Ejercicio 4 capítulo 4

```
exportaciones <- read.xlsx("Ejercicios-Cap4.xlsx", 2)
rownames(exportaciones) <- exportaciones[, 1]
exportaciones <- exportaciones[, -1]
```

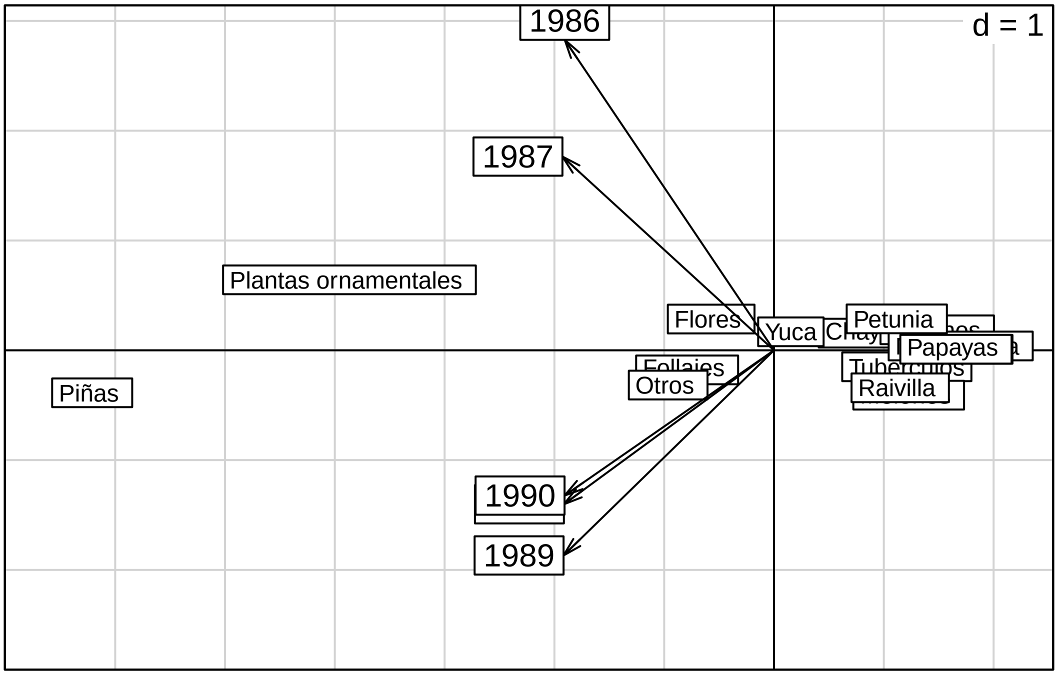
Se encuentra que sobre esta tabla de datos se puede realizar un AFC ya que a pesar de que fue ideado para describir la relación de dos variables cualitativas a través de tablas de contingencias el único requisito necesario es que las entradas de la tabla no sean negativas y que la suma de columnas y filas no se anule.

```
exportaciones_coa <- dudi.coa(df = exportaciones, scannf = F, nf = 4)
scatter(exportaciones_coa, posieig = "none")
```



NULL

```
exportaciones_pca <- dudi.pca(df = exportaciones, scannf = F, nf = 4)
scatter(exportaciones_pca, posieig = "none")
```



NULL

3 Análisis de correspondencias múltiples

```
# We load the data
data(ours)

# We show the table
ours
```

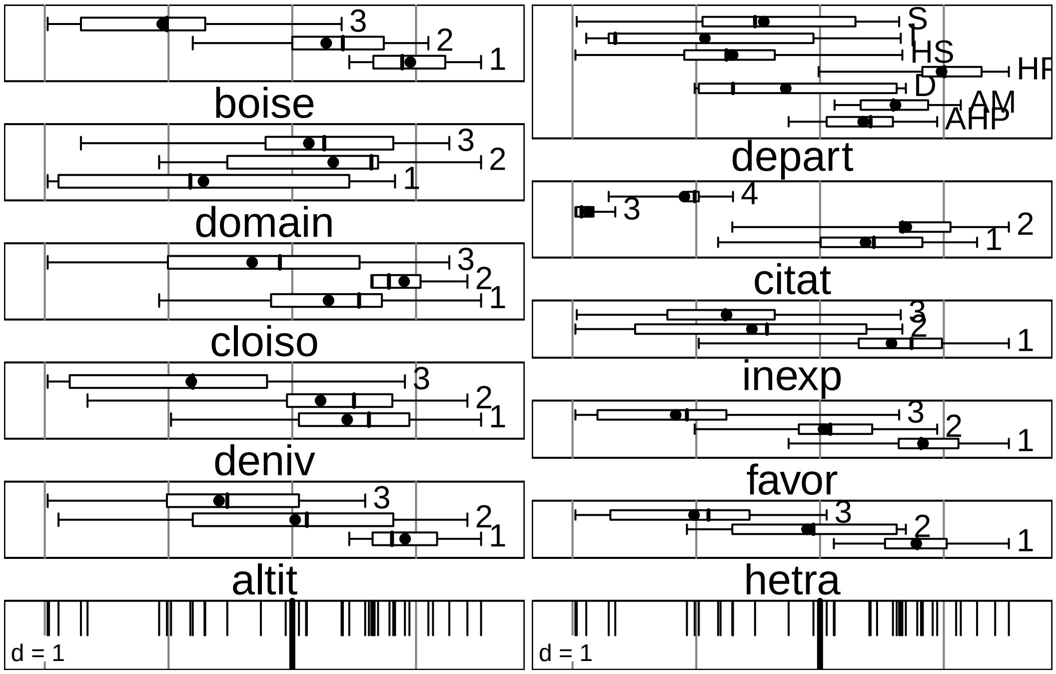
```
   altit deniv cloiso domain boise hetra favor inexp citat depart
1      2     3      3      2     2     3     3     2     1     HS
2      1     2      1      2     1     1     2     2     2     HS
3      3     3      3      2     2     2     3     3     2     HS
4      3     3      3      1     3     3     3     2     3     HS
5      3     3      1      2     2     3     2     3     1      S
6      3     3      3      1     3     3     3     3     3      S
7      2     2      3      2     2     1     2     3     1      S
8      1     1      2      2     1     1     3     2     2      S
9      2     3      1      2     3     2     3     3     4      S
```

10	2	2	3	1	3	3	2	3	1	S
11	1	1	1	1	1	1	2	2	1	S
12	2	2	3	1	3	3	3	2	3	I
13	2	3	3	1	3	3	3	2	3	I
14	1	3	2	2	1	1	1	3	2	I
15	2	2	1	3	2	2	2	2	1	I
16	3	3	3	3	3	3	3	3	4	I
17	3	1	3	3	3	3	3	1	4	D
18	3	2	3	3	3	3	2	2	4	D
19	2	1	1	3	3	3	3	1	4	D
20	2	1	1	2	2	2	1	1	2	D
21	2	1	1	2	2	2	1	1	1	D
22	1	1	1	2	1	1	1	1	2	HP
23	2	2	2	2	1	1	1	1	2	HP
24	1	1	3	3	1	1	1	1	1	HP
25	2	3	2	3	2	1	1	1	1	HP
26	2	2	1	1	2	1	1	1	1	HP
27	2	2	3	1	1	1	1	1	1	HP
28	3	2	1	3	3	3	2	1	1	HP
29	2	1	1	2	2	1	1	1	1	HP
30	1	1	3	3	1	1	2	1	1	AHP
31	3	1	3	3	2	3	2	1	1	AHP
32	3	2	3	3	2	1	2	1	1	AHP
33	3	2	3	3	2	1	1	1	1	AHP
34	3	1	3	1	3	3	1	1	1	AHP
35	1	2	3	3	1	1	1	1	1	AM
36	2	2	3	3	2	1	1	1	1	AM
37	3	1	3	3	3	1	1	1	1	AM
38	2	3	3	3	2	1	2	2	1	AM

```r
# We show a summary of the data
summary(ours)
```

altit	deniv	cloiso	domain	boise	hetra	favor	inexp	citat	depart
1: 8	1:13	1:12	1: 9	1:10	1:19	1:15	1:20	1:22	AHP:5
2:17	2:14	2: 4	2:13	2:15	2: 5	2:12	2:10	2: 7	AM :4
3:13	3:11	3:22	3:16	3:13	3:14	3:11	3: 8	3: 4	D :5
								4: 5	HP :8
									HS :4
									I :5
									S :7

```r
# We make a dispersion plot, boxplot
boxplot(dudi.acm(ours, scan = FALSE))
```




```r
# We make the disjuntive code
our_disyuntivo <- acm.disjonctif(ours)

# We make a burt table
acm.burt(ours, ours)
```

	altit.1	altit.2	altit.3	deniv.1	deniv.2	deniv.3	cloiso.1	cloiso.2
altit.1	8	0	0	5	2	1	3	2
altit.2	0	17	0	4	8	5	7	2
altit.3	0	0	13	4	4	5	2	0
deniv.1	5	4	4	13	0	0	6	1
deniv.2	2	8	4	0	14	0	4	1
deniv.3	1	5	5	0	0	11	2	2
cloiso.1	3	7	2	6	4	2	12	0
cloiso.2	2	2	0	1	1	2	0	4
cloiso.3	3	8	11	6	9	7	0	0
domain.1	1	5	3	2	4	3	2	0
domain.2	4	7	2	5	3	5	7	3

domain.3	3	5	8	6	7	3	3	1
boise.1	8	2	0	5	4	1	3	3
boise.2	0	10	5	4	6	5	6	1
boise.3	0	5	8	4	4	5	3	0
hetra.1	8	8	3	7	9	3	5	4
hetra.2	0	4	1	2	1	2	4	0
hetra.3	0	5	9	4	4	6	3	0
favor.1	4	8	3	7	6	2	5	3
favor.2	3	4	5	3	7	2	5	0
favor.3	1	5	5	3	1	7	2	1
inexp.1	4	9	7	11	8	1	7	2
inexp.2	3	5	2	2	4	4	3	1
inexp.3	1	3	4	0	2	6	2	1
citat.1	4	11	7	8	10	4	7	1
citat.2	4	2	1	3	2	2	3	3
citat.3	0	2	2	0	1	3	0	0
citat.4	0	2	3	2	1	2	2	0
depart.AHP	1	0	4	3	2	0	0	0
depart.AM	1	2	1	1	2	1	0	0
depart.D	0	3	2	4	1	0	3	0
depart.HP	2	5	1	3	4	1	4	2
depart.HS	1	1	2	0	1	3	1	0
depart.I	1	3	1	0	2	3	1	1
depart.S	2	3	2	2	2	3	3	1

	cloiso.3	domain.1	domain.2	domain.3	boise.1	boise.2	boise.3	hetra.1
altit.1	3	1	4	3	8	0	0	8
altit.2	8	5	7	5	2	10	5	8
altit.3	11	3	2	8	0	5	8	3
deniv.1	6	2	5	6	5	4	4	7
deniv.2	9	4	3	7	4	6	4	9
deniv.3	7	3	5	3	1	5	5	3
cloiso.1	0	2	7	3	3	6	3	5
cloiso.2	0	0	3	1	3	1	0	4
cloiso.3	22	7	3	12	4	8	10	10
domain.1	7	9	0	0	2	1	6	3
domain.2	3	0	13	0	5	7	1	7
domain.3	12	0	0	16	3	7	6	9
boise.1	4	2	5	3	10	0	0	10
boise.2	8	1	7	7	0	15	0	8
boise.3	10	6	1	6	0	0	13	1
hetra.1	10	3	7	9	10	8	1	19
hetra.2	1	0	4	1	0	4	1	0
hetra.3	11	6	2	6	0	3	11	0

favor.1	7	3	6	6	6	7	2	12
favor.2	7	2	3	7	3	6	3	6
favor.3	8	4	4	3	1	2	8	1
inexp.1	11	3	5	12	6	9	5	13
inexp.2	6	4	3	3	3	3	4	4
inexp.3	5	2	5	1	1	3	4	2
citat.1	14	5	5	12	5	13	4	14
citat.2	1	0	7	0	5	2	0	5
citat.3	4	4	0	0	0	0	4	0
citat.4	3	0	1	4	0	0	5	0
depart.AHP	5	1	0	4	1	3	1	3
depart.AM	4	0	0	4	1	2	1	4
depart.D	2	0	2	3	0	2	3	0
depart.HP	2	2	3	3	4	3	1	7
depart.HS	3	1	3	0	1	2	1	1
depart.I	3	2	1	2	1	1	3	1
depart.S	3	3	4	0	2	2	3	3

	hetra.2	hetra.3	favor.1	favor.2	favor.3	inexp.1	inexp.2	inexp.3
altit.1	0	0	4	3	1	4	3	1
altit.2	4	5	8	4	5	9	5	3
altit.3	1	9	3	5	5	7	2	4
deniv.1	2	4	7	3	3	11	2	0
deniv.2	1	4	6	7	1	8	4	2
deniv.3	2	6	2	2	7	1	4	6
cloiso.1	4	3	5	5	2	7	3	2
cloiso.2	0	0	3	0	1	2	1	1
cloiso.3	1	11	7	7	8	11	6	5
domain.1	0	6	3	2	4	3	4	2
domain.2	4	2	6	3	4	5	3	5
domain.3	1	6	6	7	3	12	3	1
boise.1	0	0	6	3	1	6	3	1
boise.2	4	3	7	6	2	9	3	3
boise.3	1	11	2	3	8	5	4	4
hetra.1	0	0	12	6	1	13	4	2
hetra.2	5	0	2	1	2	2	1	2
hetra.3	0	14	1	5	8	5	5	4
favor.1	2	1	15	0	0	14	0	1
favor.2	1	5	0	12	0	4	5	3
favor.3	2	8	0	0	11	2	5	4
inexp.1	2	5	14	4	2	20	0	0
inexp.2	1	5	0	5	5	0	10	0
inexp.3	2	4	1	3	4	0	0	8
citat.1	2	6	11	10	1	15	4	3

citat.2	2	0	4	1	2	3	2	2
citat.3	0	4	0	0	4	0	3	1
citat.4	1	4	0	1	4	2	1	2
depart.AHP	0	2	2	3	0	5	0	0
depart.AM	0	0	3	1	0	3	1	0
depart.D	2	3	2	1	2	4	1	0
depart.HP	0	1	7	1	0	8	0	0
depart.HS	1	2	0	1	3	0	3	1
depart.I	1	3	1	1	3	0	3	2
depart.S	1	3	0	4	3	0	2	5

	citat.1	citat.2	citat.3	citat.4	depart.AHP	depart.AM	depart.D
altit.1	4	4	0	0	1	1	0
altit.2	11	2	2	2	0	2	3
altit.3	7	1	2	3	4	1	2
deniv.1	8	3	0	2	3	1	4
deniv.2	10	2	1	1	2	2	1
deniv.3	4	2	3	2	0	1	0
cloiso.1	7	3	0	2	0	0	3
cloiso.2	1	3	0	0	0	0	0
cloiso.3	14	1	4	3	5	4	2
domain.1	5	0	4	0	1	0	0
domain.2	5	7	0	1	0	0	2
domain.3	12	0	0	4	4	4	3
boise.1	5	5	0	0	1	1	0
boise.2	13	2	0	0	3	2	2
boise.3	4	0	4	5	1	1	3
hetra.1	14	5	0	0	3	4	0
hetra.2	2	2	0	1	0	0	2
hetra.3	6	0	4	4	2	0	3
favor.1	11	4	0	0	2	3	2
favor.2	10	1	0	1	3	1	1
favor.3	1	2	4	4	0	0	2
inexp.1	15	3	0	2	5	3	4
inexp.2	4	2	3	1	0	1	1
inexp.3	3	2	1	2	0	0	0
citat.1	22	0	0	0	5	4	1
citat.2	0	7	0	0	0	0	1
citat.3	0	0	4	0	0	0	0
citat.4	0	0	0	5	0	0	3
depart.AHP	5	0	0	0	5	0	0
depart.AM	4	0	0	0	0	4	0
depart.D	1	1	0	3	0	0	5
depart.HP	6	2	0	0	0	0	0

depart.HS	1	2	1	0	0	0	0
depart.I	1	1	2	1	0	0	0
depart.S	4	1	1	1	0	0	0

	depart.HP	depart.HS	depart.I	depart.S
altit.1	2	1	1	2
altit.2	5	1	3	3
altit.3	1	2	1	2
deniv.1	3	0	0	2
deniv.2	4	1	2	2
deniv.3	1	3	3	3
cloiso.1	4	1	1	3
cloiso.2	2	0	1	1
cloiso.3	2	3	3	3
domain.1	2	1	2	3
domain.2	3	3	1	4
domain.3	3	0	2	0
boise.1	4	1	1	2
boise.2	3	2	1	2
boise.3	1	1	3	3
hetra.1	7	1	1	3
hetra.2	0	1	1	1
hetra.3	1	2	3	3
favor.1	7	0	1	0
favor.2	1	1	1	4
favor.3	0	3	3	3
inexp.1	8	0	0	0
inexp.2	0	3	3	2
inexp.3	0	1	2	5
citat.1	6	1	1	4
citat.2	2	2	1	1
citat.3	0	1	2	1
citat.4	0	0	1	1
depart.AHP	0	0	0	0
depart.AM	0	0	0	0
depart.D	0	0	0	0
depart.HP	8	0	0	0
depart.HS	0	4	0	0
depart.I	0	0	5	0
depart.S	0	0	0	7

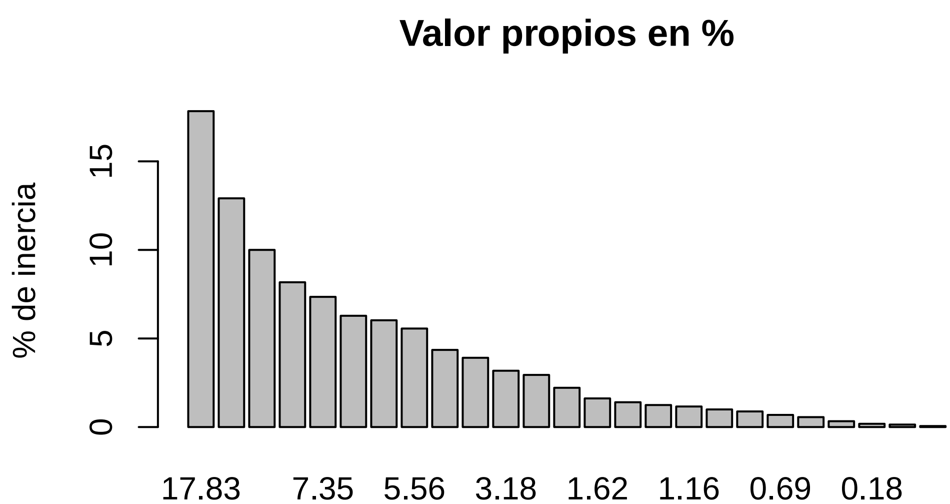
```r
# ACM results
ours_acm <- dudi.acm(ours, scann = FALSE, nf = 3)

# We check the eigen values
apply(ours_acm$cr, 2, mean)
```

```
      RS1       RS2       RS3 
0.4458097 0.3228212 0.2500239 
```

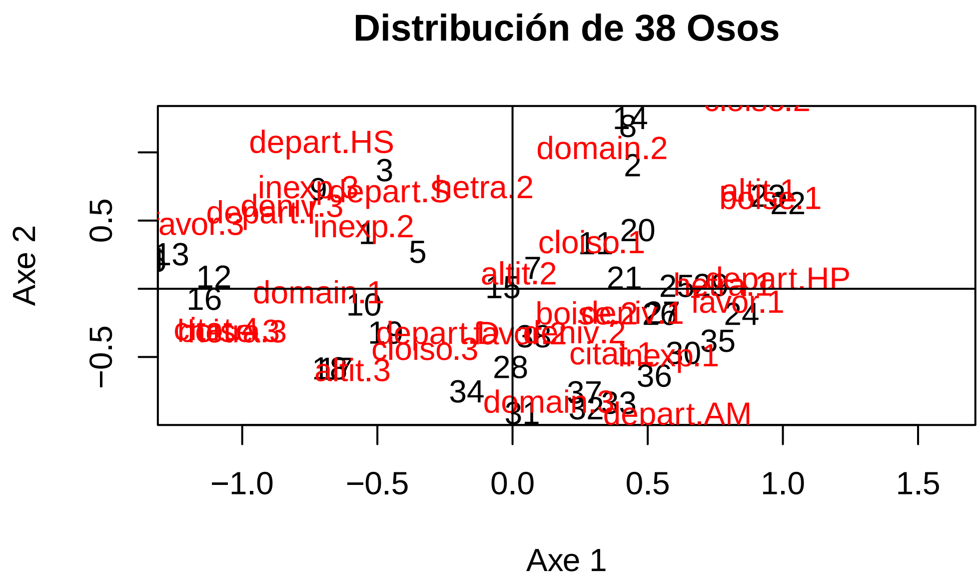
```r
ours_acm$eig[1:ours_acm$nf]
```

```
[1] 0.4458097 0.3228212 0.2500239
```

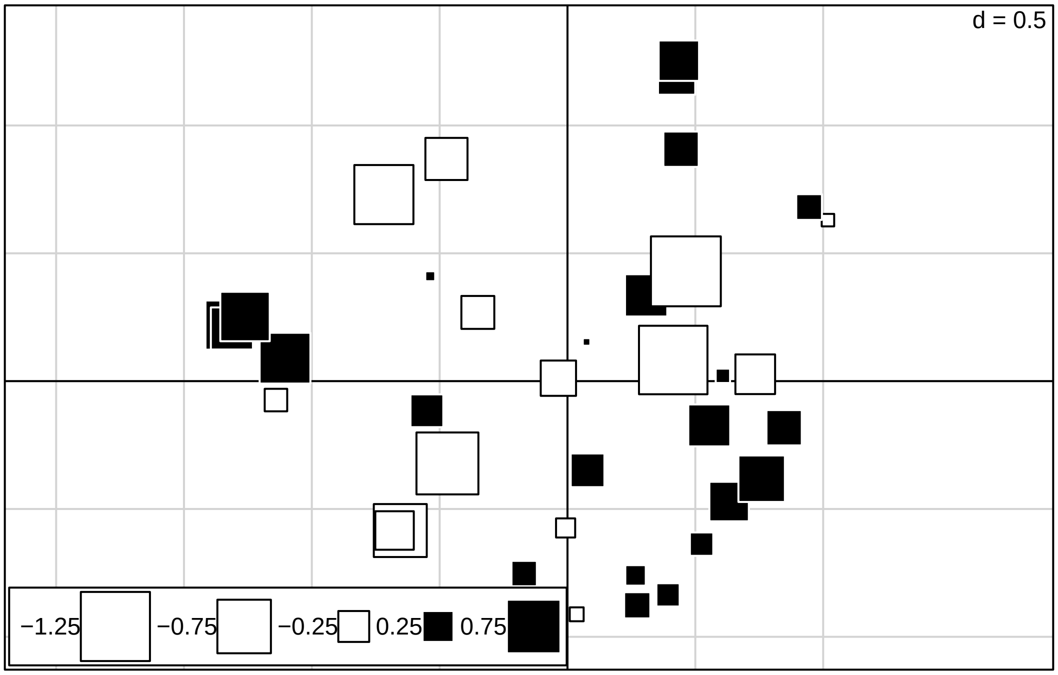
```r
# Plot of eigen values
inercia <- ours_acm$eig / sum(ours_acm$eig) * 100
barplot(inercia, ylab = "% de inercia", names.arg = round(inercia, 2))
title("Valor propios en %")
```



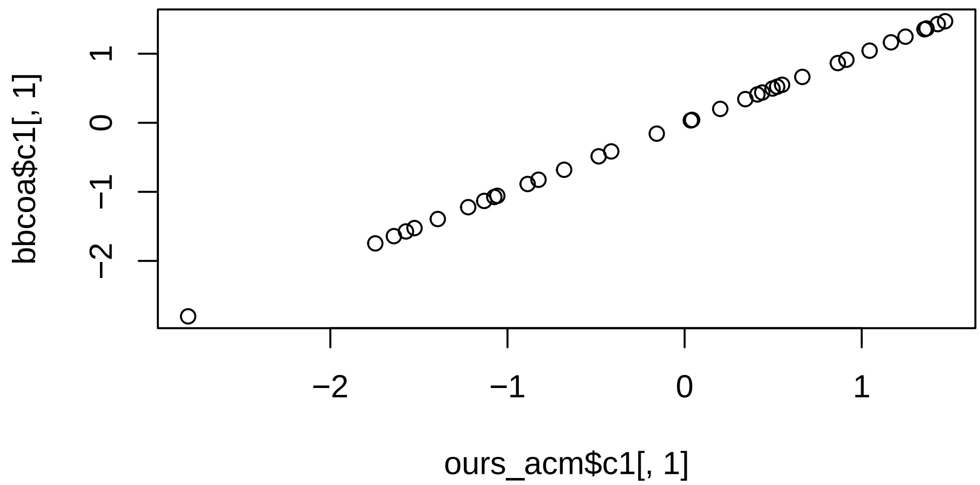
```r
# Factorial plane representation
plot(ours_acm$li[, 1], ours_acm$li[, 2], type = "n", xlab = "Axe 1", ylab = "Axe 2", xlim = c(-
text(ours_acm$li[, 1], ours_acm$li[, 2], label = row.names(ours))
text(ours_acm$co[, 1], ours_acm$co[, 2], label = colnames(our_disyuntivo), col = "red")
title("Distribución de 38 Osos")
abline(h = 0, v = 0)
```



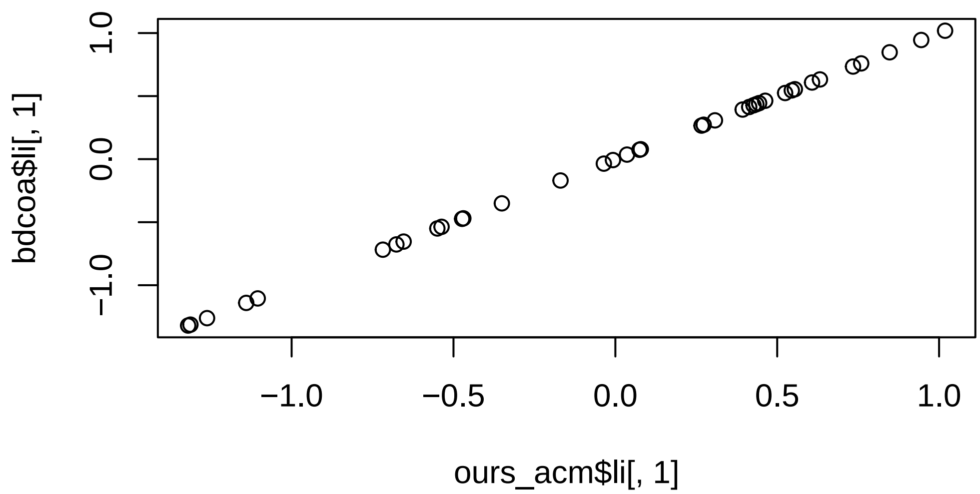

```r
# Main plane with intensities
s.value(ours_acm$li, ours_acm$li[, 3])
```



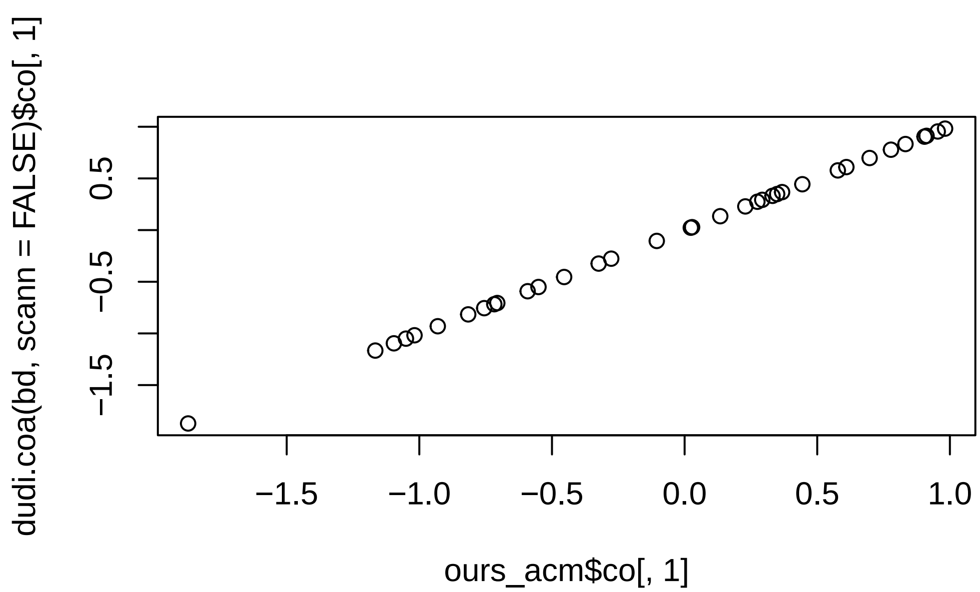

```
bb <- acm.burt(ours, ours)
bbcoa <- dudi.coa(bb, scann = FALSE)
plot(ours_acm$c1[, 1], bbcoa$c1[, 1])
```




```
bd <- acm.disjonctif(ours)
bdcoa <- dudi.coa(bd, scann = FALSE)
plot(ours_acm$li[, 1], bdcoa$li[, 1])
```




```r
plot(ours_acm$co[, 1], dudi.coa(bd, scann = FALSE)$co[, 1])
```



m)

3.1 Ejercicio 6 del capítulo 5

```
df_ejercicio_6_cap_5 <- read.xlsx("./Ejercicios-Cap5.xlsx", 1)

perros <- df_ejercicio_6_cap_5 %>%
    select(-Raza) %>%
    mutate(across(where(is.numeric), as.factor))

perros_acm <- dudi.acm(df = perros[, -7], scann = FALSE, nf = 5)
```

Se procede a ver los autovalores el grado de incercia explicada por cada uno

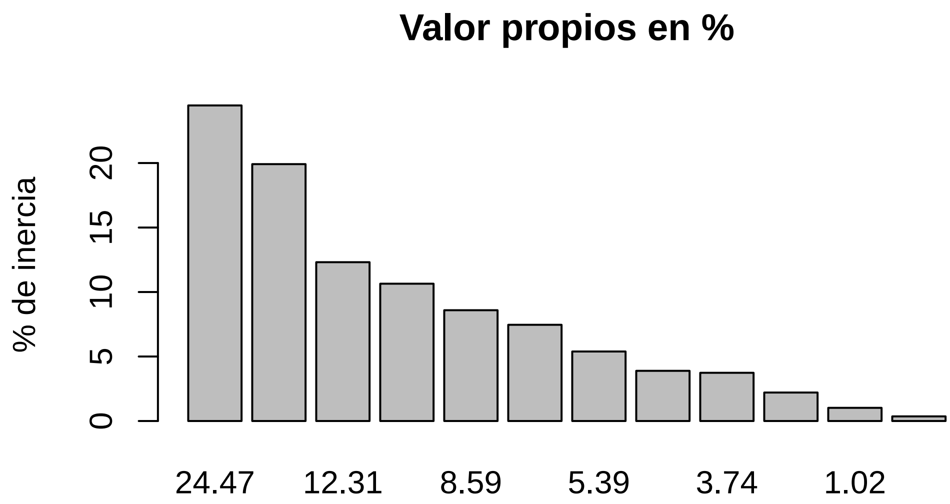
```
# We make a plot of the inertia
(inercia <- perros_acm$eig / sum(perros_acm$eig) * 100)
```

```
 [1] 24.4663585 19.9164674 12.3136391 10.6449215  8.5909715  7.4596115
 [7]  5.3903219  3.8915000  3.7371154  2.2088453  1.0205833  0.3596645
```

```
cumsum(inercia)
```

```
 [1]  24.46636  44.38283  56.69647  67.34139  75.93236  83.39197  88.78229
 [8]  92.67379  96.41091  98.61975  99.64034 100.00000
```

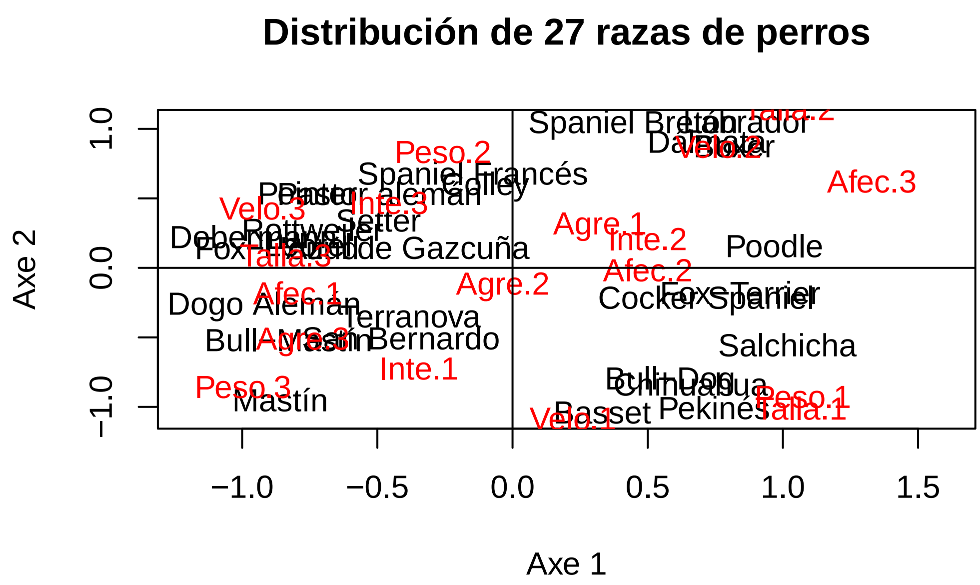
```
barplot(inercia, ylab = "% de inercia", names.arg = round(inercia, 2))
title("Valor propios en %")
```



Si se pone como regla general que se tenga una inercia mayor a 75% entonces se pueden conservar los primeros 5 ejes. Además, se tiene que los que superan 100/12 = 8.3333333 en inercia, entonces se conservaría igualmente hasta el quinto.

Graficando se obtiene

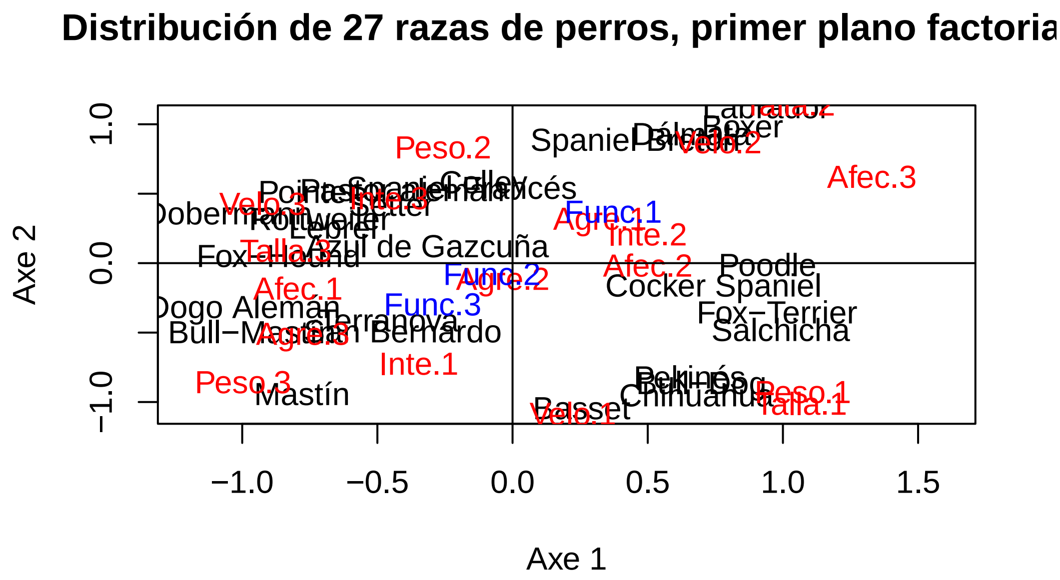
```
perros_disyuntivo <- acm.disjonctif(perros[, -7])
rownames(perros) <- df_ejercicio_6_cap_5$Raza
# Factorial plane representation
plot(perros_acm$li[, 1], perros_acm$li[, 2], type = "n", xlab = "Axe 1", ylab = "Axe 2", xlim =
text(jitter(perros_acm$li[, 1], amount = 0.1), jitter(perros_acm$li[, 2], amount = 0.1), label
# text((perros_acm$li[, 1]), (perros_acm$li[, 2]), label = row.names(perros))
text((perros_acm$co[, 1]), (perros_acm$co[, 2]), label = colnames(perros_disyuntivo), col = "re
title("Distribución de 27 razas de perros")
abline(h = 0, v = 0)
```



En este caso se agregó un poco de ruido ya que algunas razas se encontraban justamente por encima de las otras. Se puede observar que los chihuha y pekinés son colocados de forma cercana de tal forma que estos se encuentran principalmente caracterizados por ser livianos, lkentos y pequeños a ellos se contraponen los perros más grandes como es el Rottweiler por ejemplo que tiene talla velocidad y talla mayor. Se puede observar que muy pocos son los perros que son caracterizados principalmente por agresividad 1, inteligencia 2 y afecto 3.

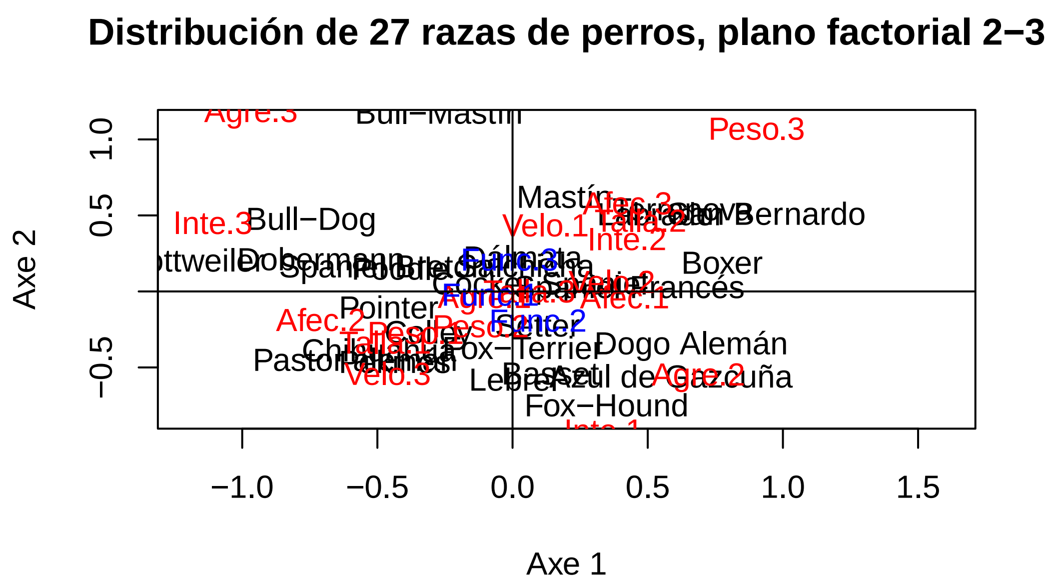
Ahora si se grafica la variable suplementaria función se tiene lo siguiente

```
funcion_suplementaria <- acm.disjonctif(perros %>% select(Func))
coord_funcion <- supcol(x = perros_acm, Xsup = funcion_suplementaria)


plot(perros_acm$li[, 1], perros_acm$li[, 2], type = "n", xlab = "Axe 1", ylab = "Axe 2", xlim =
text(jitter(perros_acm$li[, 1], amount = 0.1), jitter(perros_acm$li[, 2], amount = 0.1), label
# text((perros_acm$li[, 1]), (perros_acm$li[, 2]), label = row.names(perros))
text((perros_acm$co[, 1]), (perros_acm$co[, 2]), label = colnames(perros_disyuntivo), col = "re
text((coord_funcion$cosup[, 1]), (coord_funcion$cosup[, 1]), label = colnames(funcion_suplement
title("Distribución de 27 razas de perros, primer plano factorial")
abline(h = 0, v = 0)
```



Se puede observar en este caso que compañía y utilidad se contraponen, sin embargo, se puede observar que caza no se ve tan bien reflejada en este plano. Además, se puede obsevar que los que más aportan a utilidad son los terranova, dogo alemán y San Bernardo. Mientras que en el caso de compañía en el mismo cuadrante se encuentran algunos muy alejados. Ahora si se grafica el tercer eje con el cuarto eje se obtiene lo siguiente

```
plot(perros_acm$li[, 3], perros_acm$li[, 4], type = "n", xlab = "Axe 1", ylab = "Axe 2", xlim =
text(jitter(perros_acm$li[, 3], amount = 0.1), jitter(perros_acm$li[, 4], amount = 0.1), label
# text((perros_acm$li[, 1]), (perros_acm$li[, 2]), label = row.names(perros))
text((perros_acm$co[, 3]), (perros_acm$co[, 4]), label = colnames(perros_disyuntivo), col = "re
text((coord_funcion$cosup[, 3]), (coord_funcion$cosup[, 4]), label = colnames(funcion_suplement
title("Distribución de 27 razas de perros, plano factorial 2-3")
abline(h = 0, v = 0)
```



En este caso se puede ver más claramente la función de utilidad, que se puede observar caza se contrapone a utilidad teniendo que caza se ve mayoritariamente representado por el cuarto eje. Además, se encuentra que el que más le aporta en este caso a la posición son los Setter, perros los cuales realmente tienen como utilidad caza, lo cual termina siendo interesante ya que esta variable no fue incluida inicialmente en el ajuste del ACM.

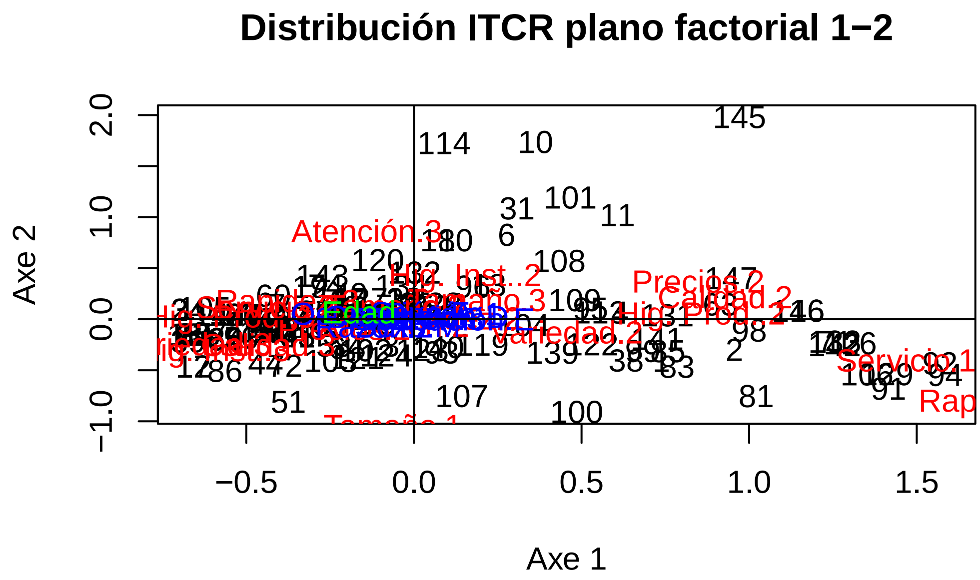
n)

```
# We read the data
library(readxl)
df_itcr <- read_excel("./ComedorITCR-datos.xls")[, -1]



# Disjonctif code of itcr table first 9 columns
itcr_disyuntivo <- acm.disjonctif(as.data.frame(df_itcr %>% select(1:9)))


itcr_acm <- dudi.acm(df = as.data.frame(df_itcr %>% select(c(1:9))) %>% mutate(across(where(is.


# We are going to get the supplementart vary variable of disjointif code for qualitative
```

```r
# variables
colnames(df_itcr)
```

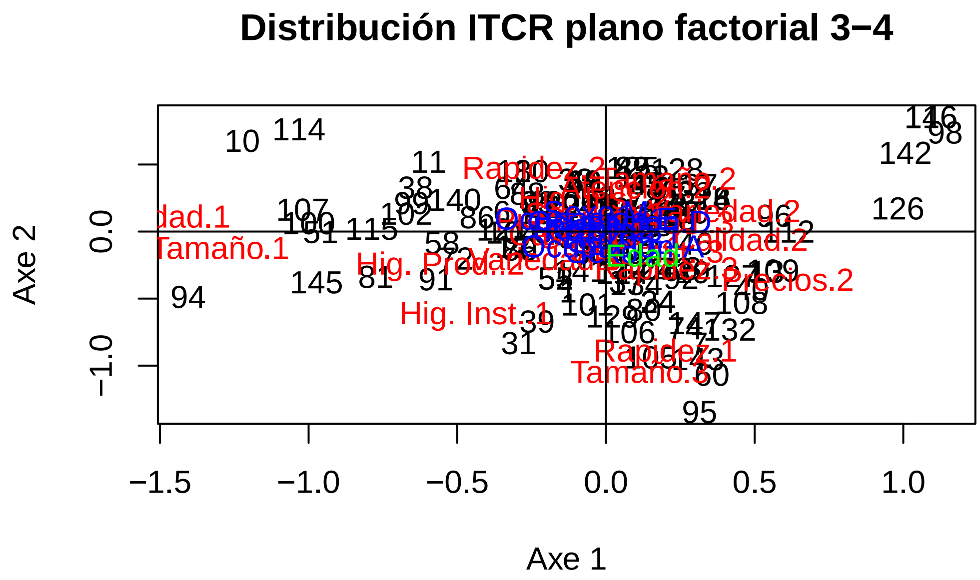
```
 [1] "Rapidez"     "Hig. Inst." "Hig. Prod." "Variedad"   "Calidad"
 [6] "Precios"     "Tamaño"     "Atención"   "Servicio"   "Razón"
[11] "Edad"        "Sexo"       "Ocupación"  "CatEdad"
```

```r
disyuntivo_suplementarias <- acm.disjonctif(as.data.frame(df_itcr %>% select(Razón, Sexo, Ocupa
coord_suplementarias <- supcol(x = itcr_acm, Xsup = disyuntivo_suplementarias)

edad_sup <- data.frame((df_itcr$Edad - mean(df_itcr$Edad)) / sd(df_itcr$Edad))
v_h <- supcol(x = itcr_acm, Xsup = edad_sup)


coord_vh_continua <- v_h$cosup # / nrow(itcr_disyuntivo)

# We plot the variables, and the supplementary variables
plot(itcr_acm$li[, 1], itcr_acm$li[, 2], type = "n", xlab = "Axe 1", ylab = "Axe 2")
text(itcr_acm$li[, 1], itcr_acm$li[, 2], label = row.names(df_itcr))
text(itcr_acm$co[, 1], itcr_acm$co[, 2], label = colnames(itcr_disyuntivo), col = "red")
text(coord_suplementarias$cosup[, 1], coord_suplementarias$cosup[, 2], label = colnames(disyunt
text(coord_vh_continua[1], coord_vh_continua[2], label = "Edad", col = "green")
title("Distribución ITCR plano factorial 1-2")
abline(h = 0, v = 0)
```



Se puede observar que en este la variable de edad no se ve la mejor forma representada. Se contraponen servicio 1 a rapidez 3, servicio 2 y tamaño 2. Además, se observar que los individuos se muestran realmente distribuidos a los largo de todos los ejes. Las variables suplementarias no se ven representadas de las mejor forma en este plano y sucede lo mismo que con edad.

```
# We plot the variables, and the supplementary variables in factorial plane 3-4
plot(itcr_acm$li[, 3], itcr_acm$li[, 4], type = "n", xlab = "Axe 1", ylab = "Axe 2")
text(itcr_acm$li[, 3], itcr_acm$li[, 4], label = row.names(df_itcr))
text(itcr_acm$co[, 3], itcr_acm$co[, 4], label = colnames(itcr_disyuntivo), col = "red")
text(coord_suplementarias$cosup[, 3], coord_suplementarias$cosup[, 4], label = colnames(disyunt
text(coord_vh_continua[3], coord_vh_continua[4], label = "Edad", col = "green")
title("Distribución ITCR plano factorial 3-4")
abline(h = 0, v = 0)
```



En este segundo plano se encuentra que en el plano factorial 3-4 se ven mejo representadas las variables suplementarias.