COMPONENTES PRINCIPALES A TRAVES DE COMANDOS

Nelson de Jesus Magaña Godinez

```
library(readxl)
zonasmad <- read_excel("zonasmad.xlsx")
View(zonasmad)
```

El comando princomp permite realizar análisis en componentes principales mediante la siguiente sintaxis:

princomp{"V1+V2+ … +Vn, cor=TRUE, scores=TRUE, data=Conjunto de datos)

las variables vl, v2, …, vn son las variables a reducir, cor=TRUE permite obtener la matriz de correlaciones entre componentes y variables (matriz factorial), seores= TRUE permite calcular las puntuaciones de las componentes y data=Conjunto de datos permite declarar el conjunto de datos que contiene las variables.

En nuestro ejemplo anterior la sintaxis vía comandos sería la siguiente:

```
componentes = princomp(~ anal + nes + ocu + ocuin + ocuser + p10 + p14
  + p65 + pd + pt + tec + tm, cor=TRUE,
  scores=TRUE, data = zonasmad)
```

```
summary(componentes)
```

```
Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
Standard deviation     2.3654833 1.9943178 1.06684949 0.73590291 0.57736255
Proportion of Variance 0.4662926 0.3314420 0.09484732 0.04512942 0.02777896
Cumulative Proportion  0.4662926 0.7977346 0.89258188 0.93771130 0.96549026
                           Comp.6     Comp.7      Comp.8      Comp.9
Standard deviation     0.40504788 0.34727269 0.286548613 0.163612772
Proportion of Variance 0.01367198 0.01004986 0.006842509 0.002230762
```

```
Cumulative Proportion  0.97916224 0.98921210 0.996054613 0.998285374
                           Comp.10      Comp.11      Comp.12
Standard deviation     0.116792679 0.0772544272 3.109234e-02
Proportion of Variance 0.001136711 0.0004973539 8.056112e-05
Cumulative Proportion  0.999422085 0.9999194389 1.000000e+00
```

Para ver las puntuaciones utilizamos la sintaxis siguiente:

```
componentes$scores
```

```
        Comp.1      Comp.2       Comp.3      Comp.4       Comp.5      Comp.6
1  -1.05969437 -0.50000941  2.612448359 -0.86714912  0.338420415  0.31684912
2  -1.96336433  1.30351245  0.076390246 -1.67519254 -0.527201984 -0.18620360
3  -3.29074504  0.20831545 -0.299843625  0.47206990 -0.009291024 -0.01395031
5  -2.94605518 -1.83311133 -0.725973146  1.21884785 -0.361392394  0.41015760
6  -0.93989040 -0.08928869  0.822802976 -0.02932030  0.086712503  0.11106099
7  -2.32275744 -3.70075773  1.297724557  0.30747759  0.208838015  0.01853895
8   0.20805508 -1.10703792 -1.259277703 -1.35206163 -1.068426888 -0.24099967
9  -3.70217811  1.14646984 -0.224371936  0.58700467 -0.169812958 -0.38780277
10  4.46200808 -3.06028432 -0.007891359  0.37401786 -0.895828943  0.04159473
11  3.97505160 -1.26703033  0.498945954  0.15736507 -0.039456267  0.16070886
12  2.57614245  1.92111261  0.131786711  0.43794385  0.313211507 -0.10768211
13  1.68928065  3.02165641  0.015187175  0.51703790 -0.181402652 -0.05201383
14  2.03904900  2.24227349  1.120191355  0.14381504  0.136658760  0.10642069
15  0.41282960 -0.49225680 -2.120684146 -0.82000474  1.147137479  0.95309171
16  1.03087118 -2.08004742 -0.739030619 -0.13463067  1.248495801 -1.14095889
17 -0.12846373  3.45517626 -0.041986047  0.08945237  0.071246869 -0.01707634
18 -0.04013903  0.83130746 -1.156418753  0.57332691 -0.297908240  0.02826486
        Comp.7       Comp.8      Comp.9       Comp.10     Comp.11       Comp.12
1  -0.22476530  0.270212617 -0.22541309  0.0006097079 -0.07027374 -0.0013693258
2  -0.25800335 -0.290317665  0.03729729  0.0663404873  0.05035472  0.0493925826
3  -0.23503564  0.134509120 -0.27728515  0.0479692524 -0.02535887  0.0030850928
5   0.33452264 -0.252985907 -0.09625989  0.2259841768  0.03718263  0.0215409619
6  -0.09484753  0.057630259  0.17758887 -0.0116783464  0.14139309  0.0200517010
7   0.38597777 -0.158401819  0.13146060 -0.2210934055  0.06151567 -0.0274797732
8   0.59996221 -0.007039912 -0.07158938 -0.0440983269 -0.03736676 -0.0375965981
9  -0.25908471  0.045177990  0.34818603 -0.0367223864 -0.18802203  0.0007514015
10 -0.52391691  0.298339057  0.11396317  0.0083949563  0.01918992  0.0275476733
11 -0.19680259 -0.392231160  0.03363984  0.1188179680 -0.08106305 -0.0377715124
12  0.05781760 -0.730985534 -0.13907642 -0.1756933719 -0.03294965  0.0385289016
13  0.48140262  0.434229863  0.06518078 -0.0742299627  0.04545406  0.0325464590
14  0.62276365  0.218159457  0.01702794  0.1395395838 -0.05823785 -0.0055298411
```

```
15 -0.01828799  0.110358033  0.12820294 -0.0361909053 -0.03093582  0.0074059398
16  0.02643400  0.144504187 -0.06412205  0.1011143728  0.02998240  0.0052916548
17 -0.35108310 -0.130990175  0.09364210  0.0764007603  0.12621426 -0.0752105288
18 -0.34705337  0.249831589 -0.27244357 -0.1854645607  0.01292104 -0.0211847887
```

Para ver la matriz factorial utilizamos la siguiente sintaxis:

```
print(unclass(loadings(componentes)))
```

```
             Comp.1       Comp.2       Comp.3       Comp.4      Comp.5
anal    0.340807240  0.228799199  0.199259354  0.164176438  0.13677537
nes    -0.242558100 -0.370833813  0.107305418  0.330072333 -0.12465240
ocu     0.234775703 -0.267303092 -0.232366286 -0.742995527 -0.29245129
ocuin   0.389575793 -0.002498936 -0.199655838  0.071173335  0.46455218
ocuser  0.201871354 -0.414852559 -0.009288395 -0.138615448  0.27854390
p10     0.329783051 -0.267277537  0.212978769  0.173740598  0.05384015
p14     0.368013253 -0.133465022 -0.263892535  0.188477169 -0.28356793
p65    -0.004197741 -0.288672791  0.738022614 -0.228876256  0.21083637
pd     -0.176849828 -0.382363301 -0.372943594  0.263237138  0.26360581
pt      0.318949118 -0.207978091  0.179372685  0.313181241 -0.57833662
tec    -0.177029578 -0.443808108 -0.141208484  0.008770419  0.08995554
tm      0.410880654  0.086999916 -0.062932606  0.047213705  0.21683247
             Comp.6      Comp.7       Comp.8      Comp.9      Comp.10
anal    0.059206591  0.75621488  0.259331733  0.15900550  0.008030855
nes    -0.433647228  0.14418023 -0.252832611  0.51938446 -0.096316641
ocu    -0.132863820  0.31410020 -0.204351296  0.02905156  0.115281158
ocuin   0.007589883 -0.15741588 -0.567188801 -0.06654251 -0.270244643
ocuser  0.305734119 -0.32070525  0.472912565  0.40677329  0.165363530
p10    -0.432810654 -0.11873090 -0.040264906 -0.12311087  0.552247716
p14    -0.373033263 -0.11296929  0.448540888 -0.28648663 -0.458107797
p65    -0.022346000  0.03886158 -0.008605415 -0.36704733 -0.293894137
pd      0.150501302  0.33067547  0.008448784 -0.46520700  0.294598691
pt      0.560592637 -0.03665210 -0.263494786 -0.04609295  0.035595203
tec     0.182644612  0.20109781  0.021887274  0.12337940 -0.426723367
tm      0.002980961  0.03951312 -0.094072629  0.26172777 -0.075028684
           Comp.11     Comp.12
anal    0.08532309  0.26625002
nes    -0.31229789  0.13534462
ocu    -0.08243327  0.05352625
ocuin   0.04441334  0.40188868
ocuser -0.15473429  0.23137477
p10     0.45558681 -0.10307682
```

```
p14    -0.11274276  0.05853696
p65    -0.21846896 -0.06501455
pd     -0.32274751 -0.07220473
pt     -0.04017246  0.01561707
tec     0.64434404 -0.23835510
tm     -0.27114855 -0.78251520
```

Para aislar las puntuaciones de las tres primeras componentes usamos la sintaxis siguiente:

```
COMP1=componentes$scores[,1]
COMP2=componentes$scores[,2]
COMP3=componentes$scores[,3]
```

Ahora ya podemos representar las puntuaciones de la primera componente contra las puntuaciones de la segunda.

```
v = c("Centro", "Arganzuela", "Retiro", "Salamanca", "Chamartin", "Tetuan", "Chamberi",
      "Fuencarral", "Moncloa", "Latina",
      "Carabanchel", "Villaverde", "Mediodia",
      "Vallecas", "Moratalaz", "Ciudad Lineal",
      "San Blas", "Hortaleza")
v=as.factor(v)
```

```
library(car)
```

```
Loading required package: carData
```

```
scatterplot(COMP1 ~ COMP2, reg.line = FALSE,
            smooth = FALSE, spread = FALSE,
            boxplots = FALSE, span = 0.5,
            data = zonasmad,
            id.n = 18)
```

```
Warning in plot.window(...): "reg.line" is not a graphical parameter
```

```
Warning in plot.window(...): "spread" is not a graphical parameter
```

```
Warning in plot.window(...): "span" is not a graphical parameter
```

```
Warning in plot.window(...): "id.n" is not a graphical parameter

Warning in plot.xy(xy, type, ...): "reg.line" is not a graphical parameter

Warning in plot.xy(xy, type, ...): "spread" is not a graphical parameter

Warning in plot.xy(xy, type, ...): "span" is not a graphical parameter

Warning in plot.xy(xy, type, ...): "id.n" is not a graphical parameter

Warning in axis(side = side, at = at, labels = labels, ...): "reg.line" is not a
graphical parameter

Warning in axis(side = side, at = at, labels = labels, ...): "spread" is not a
graphical parameter

Warning in axis(side = side, at = at, labels = labels, ...): "span" is not a
graphical parameter

Warning in axis(side = side, at = at, labels = labels, ...): "id.n" is not a
graphical parameter

Warning in axis(side = side, at = at, labels = labels, ...): "reg.line" is not a
graphical parameter

Warning in axis(side = side, at = at, labels = labels, ...): "spread" is not a
graphical parameter

Warning in axis(side = side, at = at, labels = labels, ...): "span" is not a
graphical parameter

Warning in axis(side = side, at = at, labels = labels, ...): "id.n" is not a
graphical parameter

Warning in box(...): "reg.line" is not a graphical parameter

Warning in box(...): "spread" is not a graphical parameter
```

```
Warning in box(...): "span" is not a graphical parameter

Warning in box(...): "id.n" is not a graphical parameter

Warning in title(...): "reg.line" is not a graphical parameter

Warning in title(...): "spread" is not a graphical parameter

Warning in title(...): "span" is not a graphical parameter

Warning in title(...): "id.n" is not a graphical parameter
```