

Componentes Principales 2

Carlos David Contreras Chacon

2022-10-19

Parte A

```
x = read.csv("países_mundo.csv")

cat("Matrices de varianza-covarianza S\n")

## Matrices de varianza-covarianza S

S = cov(X)
S

##
##      CrecPobl      MortInf      PorcMujeres      PNB95      ProdElec
## CrecPobl      1.538298e+00      2.195026e+01      -6.498020e+00      -8.933379e+04
## MortInf      2.195026e+01      1.032859e+03      -9.249342e+00      -2.269332e+06
## PorcMujeres    -6.498020e+00      -9.249342e+00      7.668322e+01      2.613114e+05
## PNB95          -8.933379e+04      -2.269332e+06      2.613114e+05      4.999796e+10
## ProdElec      -4.973964e+04      -1.043435e+06      2.266248e+05      2.247791e+10
## LinTelf      -1.369079e+02      -4.381366e+03      4.499750e+02      2.839550e+07
## ConsAqua      -4.327092e+01      1.289211e+03      -1.568313e+03      1.097451e+07
## PropBosq      -3.687018e+00      -1.466516e+01      6.517895e+01      2.474311e+05
## PropDefor     -3.361974e-01      1.276296e+01      2.680592e-01      -5.866203e-04
## ConsEner      -8.384169e+02      -4.442568e+04      2.855207e+02      1.415626e+08
## EmisCO2       -1.137877e+00      9.485590e+01      2.159332e+00      2.563073e+05
##
##      ProdElec      LinTelf      ConsAqua      PropBosq
## CrecPobl      -4.973964e+04      -1.369079e+02      -4.327092e+01      -3.687018
## MortInf      -1.043435e+06      4.381366e+03      1.289211e+03      -1.568313
## PorcMujeres    2.266248e+05      4.499750e+02      -1.568313e+03      65.178947
## PNB95          2.247791e+10      2.839550e+07      1.097451e+07      247431.122807
## ProdElec      1.841908e+18      7.583050e+06      1.399817e+07      78359.785965
## LinTelf      7.583050e+06      3.841247e+04      1.353110e+04      248.735789
## ConsAqua      1.399817e+07      1.893110e+04      3.381981e+05      -2220.757895
## PropBosq      7.035978e+04      2.487158e+02      -2.220758e+03      401.003509
## PropDefor     -2.386560e+04      9.940401e+01      6.742972e+01      2.625635
## ConsEner      6.881296e+07      3.426262e+05      2.892242e+05      -5153.438596
## EmisCO2       1.392779e+05      6.385780e+02      4.869228e+02      -12.897193
##
##      ProdDefor      ConsEner      EmisCO2
## CrecPobl      1.392779e+05      6.385780e+02      -1.137077
## MortInf      1.276296e+01      -4.442568e+04      -94.855800
## PorcMujeres    2.688952e-01      2.855207e+02      -2.159132
## PNB95          5.866203e-04      1.415030e+08      259167.323500
## ProdElec      -3.189348e+04      6.881296e+07      139277.888640
## LinTelf      -0.948461e+01      3.426262e+05      638.570800
## ConsAqua      -6.743793e-01      2.855207e+05      496.332763
## PropBosq      2.652563e+00      5.153439e+03      -12.897193
## PropDefor     1.817253e+00      -1.051522e+03      -2.632487
## ConsEner      -1.051522e+03      5.814955e+06      10286.195781
## EmisCO2       -2.632487e+06      1.828916e+04      27.268614

cat("Matriz de correlaciones\n")

## Matriz de correlaciones

R = cor(X)
R

##
##      CrecPobl      MortInf      PorcMujeres      PNB95      ProdElec
## CrecPobl      1.00000000      0.55087948      0.55857711      0.32212154      0.29711119
## MortInf      0.55087948      1.00000000      -0.03286139      -0.31579250      -0.24053689
## PorcMujeres    -0.55857711      -0.03286139      1.00000000      0.14338826      0.19085114
## PNB95          -0.32212154      -0.31579250      0.14338826      1.00000000      0.74476001
## ProdElec      -0.29711119      -0.24053689      -0.19085114      0.74476001      1.00000000
## LinTelf      -0.56322228      -0.69558922      0.26167018      0.46539599      0.28664508
## ConsAqua      -0.06772993      -0.60975563      -0.31106243      0.08541500      0.13047653
## PropBosq      -0.18456028      -0.02778415      0.37096094      0.05205919      0.02603078
## PropDefor     0.20107881      0.29459348      0.02266339      -0.19262327      -0.17478434
## ConsEner      -0.30187731      0.61731132      0.01453216      0.28272492      0.22501894
## EmisCO2       -0.17568600      -0.56520778      0.04992837      0.21425123      0.13768017
##
##      LinTelf      ConsAqua      PropBosq      PropDefor      ConsEner
## CrecPobl      -0.56322228      -0.06772993      -0.18456028      0.20107881      -0.30187731
## MortInf      -0.69558922      -0.60975563      -0.02778415      0.29459348      0.13047653
## PorcMujeres    -0.26167018      -0.31106243      0.37096094      0.02266339      0.01453216
## PNB95          0.46539599      0.08541500      0.05205919      -0.19262327      0.28272492
## ProdElec      0.28664508      0.13047653      0.02603078      -0.17478434      0.22501894
## LinTelf      0.00000000      0.10539304      0.05205919      -0.37623901      0.70680385
## ConsAqua      0.10539304      1.00000000      0.19299225      0.08795811      0.16259804
## PropBosq      0.06337138      -0.19299225      1.00000000      0.09725032      -0.11492480
## PropDefor     -0.37623901      0.08795811      0.09725032      1.00000000      0.34833826
## ConsEner      0.70680385      0.16259804      -0.11492480      0.34833826      1.00000000
## EmisCO2       0.62393719      0.16227447      -0.12333592      -0.37396154      0.87965517
##
##      EmisCO2
## CrecPobl      -0.17568600
## MortInf      -0.56520778
## PorcMujeres    -0.04692837
## PNB95          0.22426123
## ProdElec      0.13768017
## LinTelf      0.62393719
## ConsAqua      -0.16227447
## PropBosq      -0.12333592
## PropDefor     -0.37396154
## ConsEner      0.87965517
## EmisCO2       1.00000000

eigenS = eigen(S)

eigenR = eigen(R)

lambdas = eigenS$values
totvar = sum(diag(S))
cat("Proporción de varianza explicada por cada componente\n")

## Proporción de varianza explicada por cada componente

varXcomp = lambdas / totvar
varXcomp

## [1] 0.934543e-01 9.647298e-02 6.795804e-05 4.554567e-06 1.782422e-07
## [6] 7.538917e-09 5.317786e-09 6.657763e-10 5.502887e-11 2.107843e-11
## [13] 6.808939e-12

cumsun(varXcomp)

## [1] 0.9034543 0.9992773 0.9999953 0.9999998 1.0000000 1.0000000 1.0000000
## [8] 1.0000000 1.0000000 1.0000000 1.0000000
```

La primera componente corresponde en gran medida a la varianza total explicada, la segunda tambien aporta una parte pero a partir de ahí es mínima la varianza que van aportando cada componente.

```
lambdas = eigenR$values
totvar = sum(diag(R))
cat("Proporción de varianza explicada por cada componente\n")

## Proporción de varianza explicada por cada componente

varXcomp = lambdas / totvar
varXcomp

## [1] 0.3663526 0.3745301 0.1245803 0.0789200 0.0721489 0.06629996
## [7] 0.0538368 0.0297891 0.0152789 0.0133925 0.0133925 0.00630532

cumsun(varXcomp)

## [1] 0.3663526 0.5418065 0.6663893 0.7448816 0.8171762 0.8834671 0.9354048
## [8] 0.9651132 0.9863921 0.9936947 1.0000000
```

En este caso el aumento va siendo mucho mas gradual, con mas componentes que aportan partes significantes para la varianza total de R.

Parte B

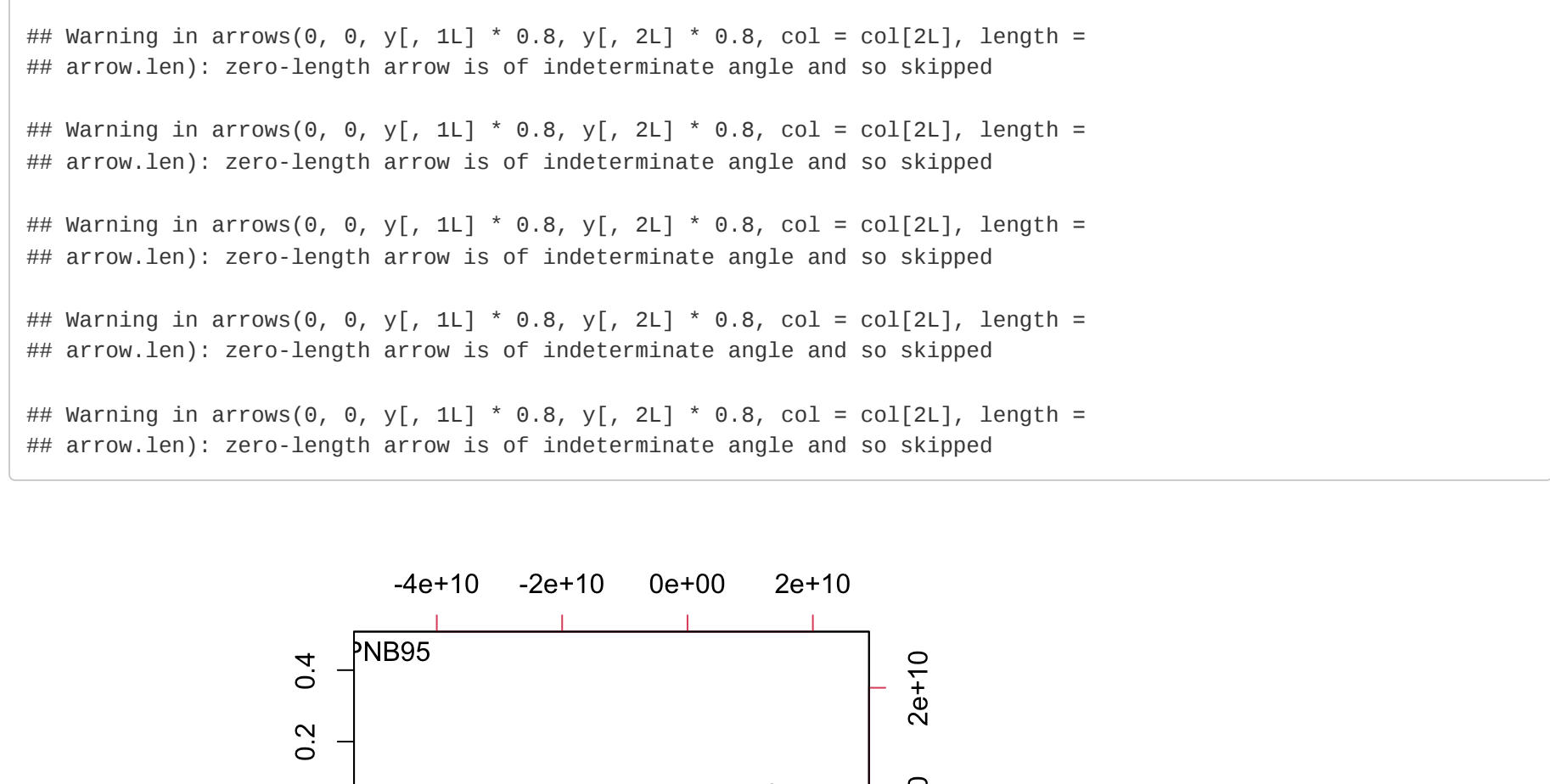
```
library(stats)
library(factoextra)

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://go.gi/vc3mBa

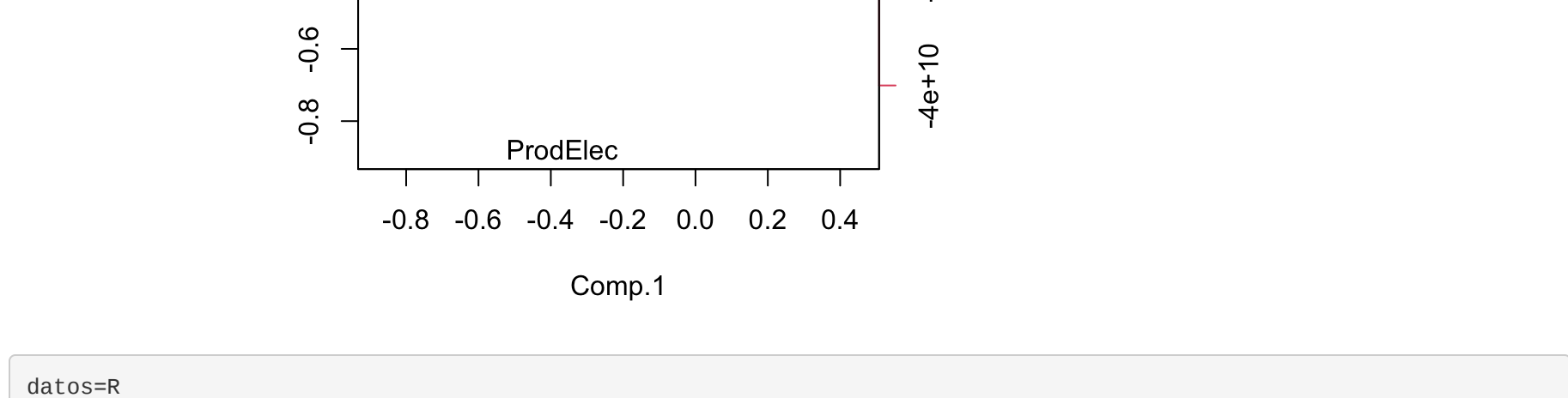
library(ggplot2)

datos=$
cp$princomp(datos,cor=FALSE)
cpas$w.mat$diag[cp$loadings]
plot(cpas[,1:2],type="p", main = "Titulo")
text(cpas[,1],cpas[,2],i:nrow(cpas))
```

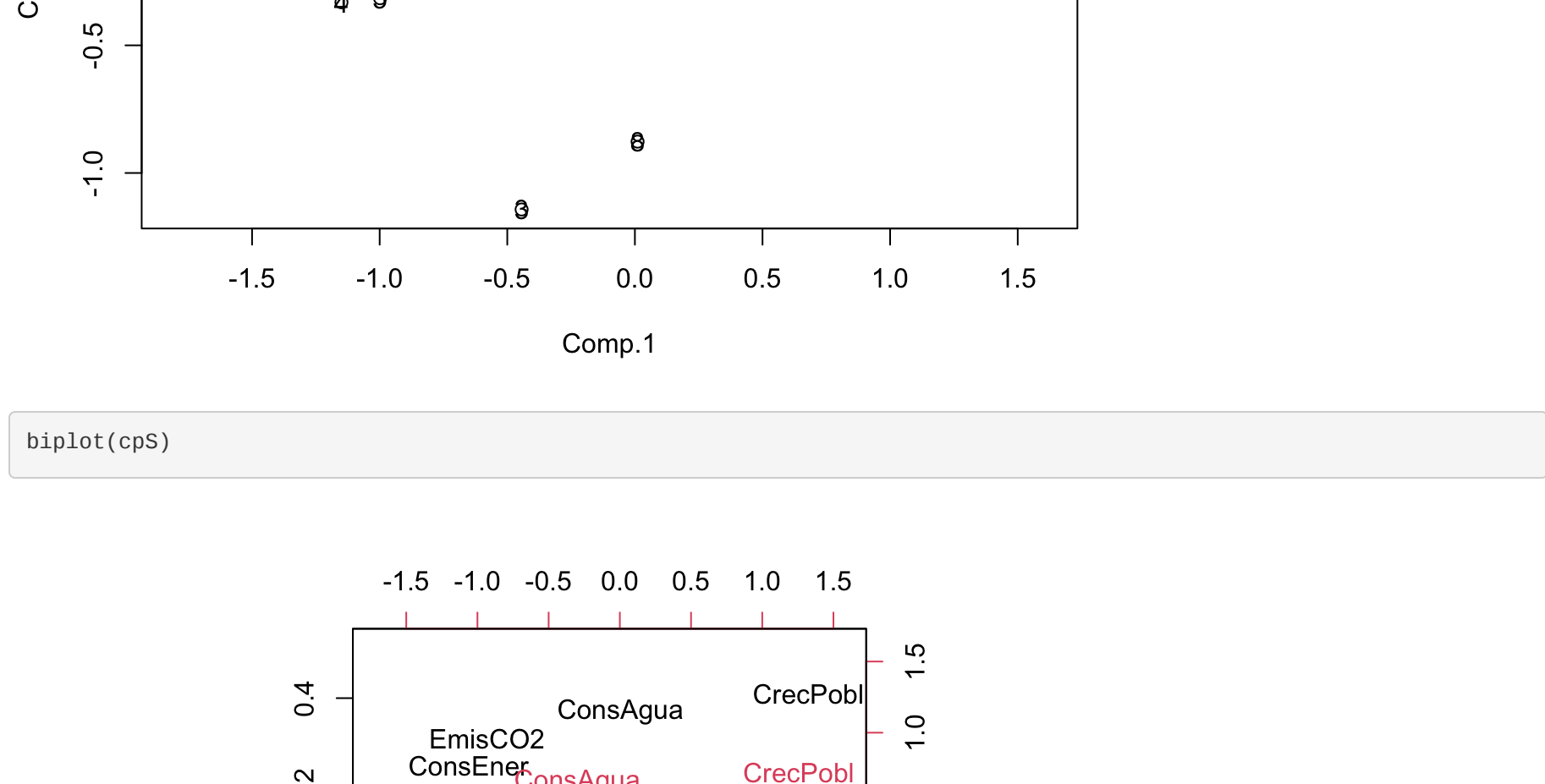


```
biplot(cps)

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
```

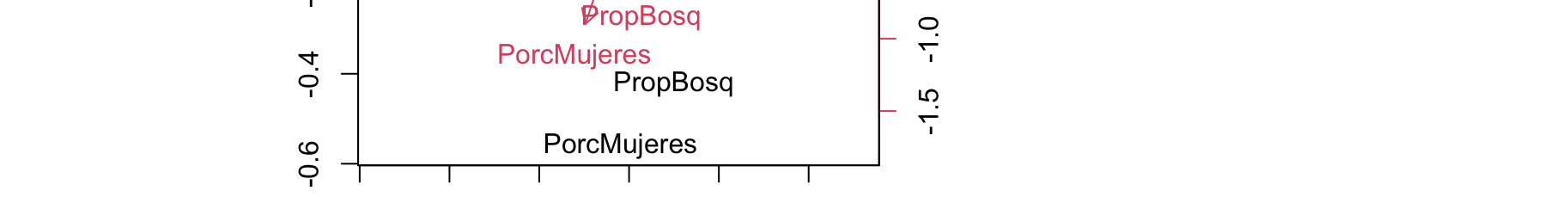


```
datos=R
cp$princomp(datos,cor=FALSE)
cpas$w.mat$diag[cp$loadings]
plot(cpas[,1:2],type="p", main = "Titulo")
text(cpas[,1],cpas[,2],i:nrow(cpas))
```



```
biplot(cps)

## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L], length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped
```

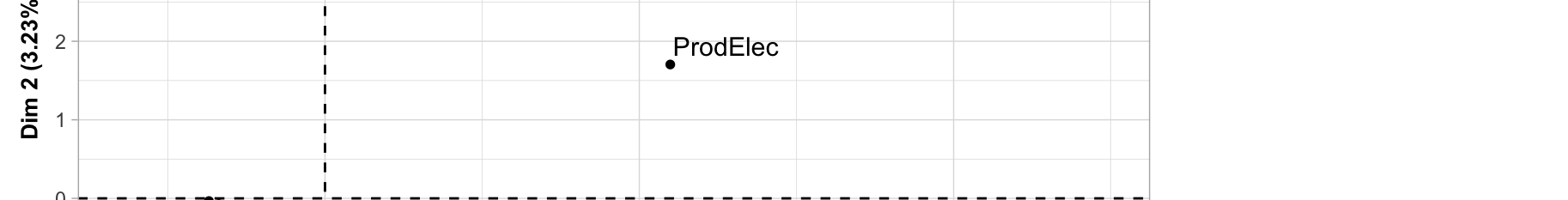
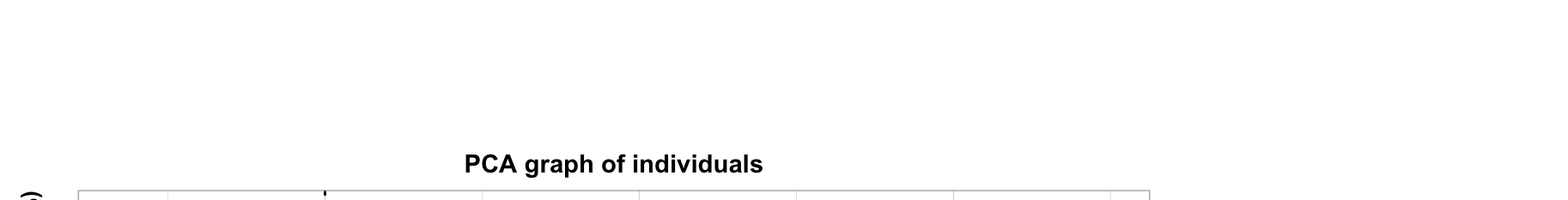


Parte C

```
library(FactoMineR)

datos=S
cp3 = PCA(datos)

## Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



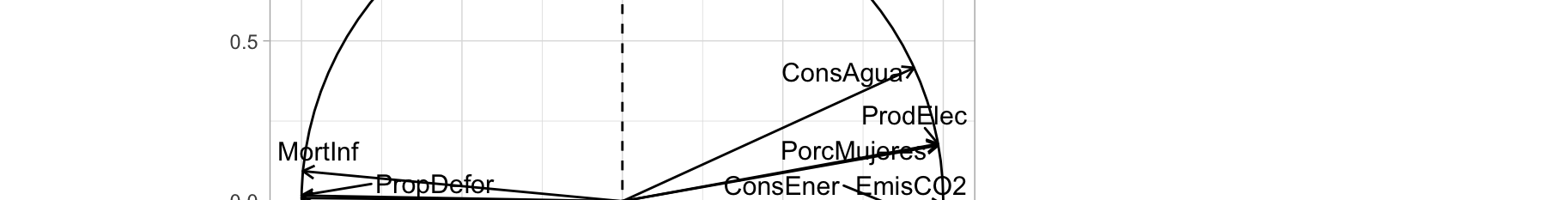
```
fviz_pca_ind(cps, col.ind = "blue", addEllipses = TRUE, repel = TRUE)

## Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

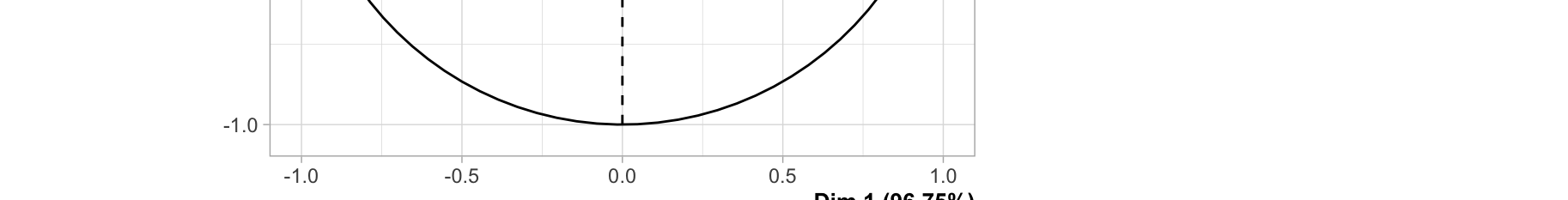


```
fviz_screplot(cps)

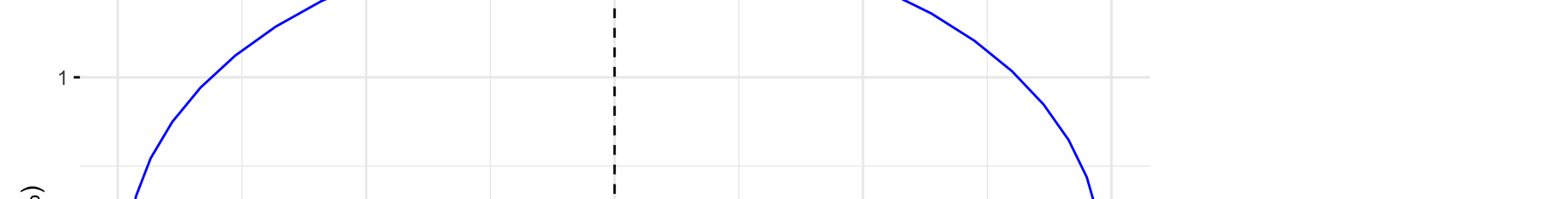
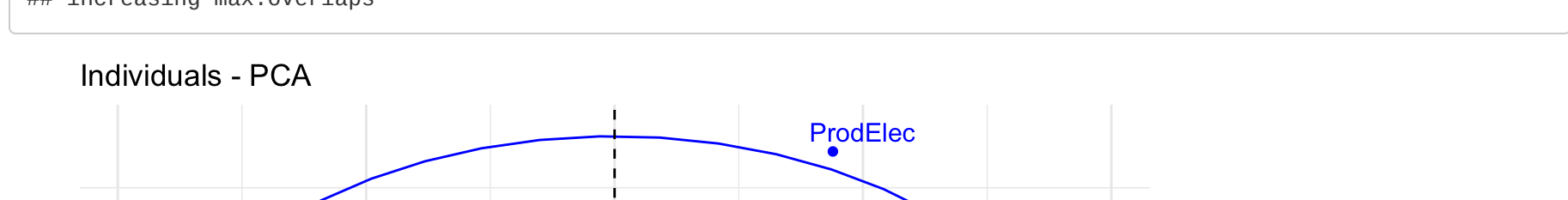
## Warning: ggplot2: 9 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
fviz_contrib(cps, choice = c("var"))
```



```
datos=R
cp3 = PCA(datos)
```

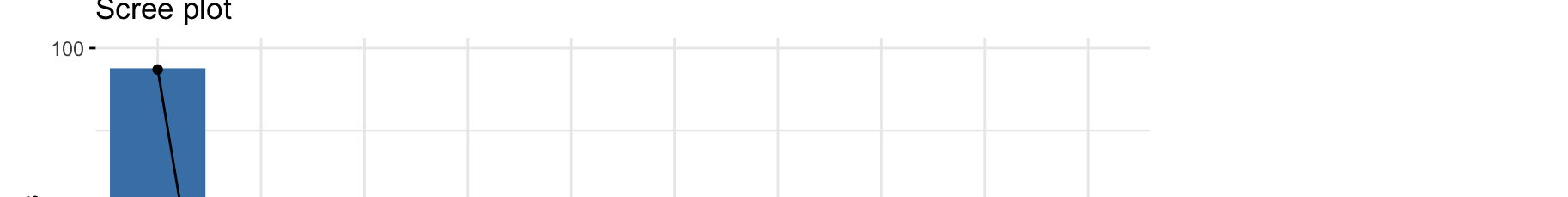


```
fviz_pca_ind(cps, col.ind = "blue", addEllipses = TRUE, repel = TRUE)
```



```
fviz_screplot(cps)

## Warning: ggplot2: 9 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
fviz_contrib(cps, choice = c("var"))
```

