Act2_Componentes Principales

Ana Lucía Cárdenas Pérez A01284090

2023-09-28

```
data <- read.csv("paises_mundo.csv")
```

1. Calcule las matrices de varianza-covarianza S con cov(X) y la matriz de correlaciones con cor(X)

```
# Matriz de Covarianza
matrizCov <- cov(data)

# Matriz de correlaciones
matrizCorr <- cor(data)

print("Matriz Covarianza:")
```

```
## [1] "Matriz Covarianza:"
```

```
matrizCov
```

```
##                 CrecPobl       MortInf   PorcMujeres         PNB95
## CrecPobl     1.538298e+00  2.195026e+01 -6.078026e+00 -8.933379e+04
## MortInf      2.195026e+01  1.032859e+03 -9.249342e+00 -2.269332e+06
## PorcMujeres -6.078026e+00 -9.249342e+00  7.698322e+01  2.813114e+05
## PNB95       -8.933379e+04 -2.269332e+06  2.813114e+05  4.999786e+10
## ProdElec    -4.973964e+04 -1.043435e+06  2.260248e+05  2.247791e+10
## LinTelf     -1.369079e+02 -4.381366e+03  4.499750e+02  2.039550e+07
## ConsAgua    -4.827092e+01 -1.288211e+03 -1.568313e+03  1.097481e+07
## PropBosq    -3.887018e+00 -1.466316e+01  6.517895e+01  2.474311e+05
## PropDefor    3.361974e-01  1.276296e+01  2.680592e-01 -5.806203e+04
## ConsEner    -8.384169e+02 -4.442568e+04  2.855207e+02  1.415628e+08
## EmisCO2     -1.137877e+00 -9.485500e+01 -2.150132e+00  2.501673e+05
##                 ProdElec       LinTelf      ConsAgua      PropBosq
## CrecPobl    -4.973964e+04 -1.369079e+02 -4.827092e+01     -3.887018
## MortInf     -1.043435e+06 -4.381366e+03 -1.288211e+03    -14.663158
## PorcMujeres  2.260248e+05  4.499750e+02 -1.568313e+03     65.178947
## PNB95        2.247791e+10  2.039550e+07  1.097481e+07 247431.122807
## ProdElec     1.821909e+10  7.583050e+06  1.399817e+07  70359.785965
## LinTelf      7.583050e+06  3.841247e+04  1.193110e+04    248.715789
## ConsAgua     1.399817e+07  1.193110e+04  3.301981e+05  -2220.757895
## PropBosq     7.035979e+04  2.487158e+02 -2.220758e+03    401.003509
## PropDefor   -3.180340e+04 -9.940461e+01 -6.743793e+01      2.625263
## ConsEner     6.801296e+07  3.426262e+05  2.092242e+05  -5153.438596
## EmisCO2      1.392779e+05  6.385700e+02  4.869328e+02    -12.897193
```

```
##                 PropDefor      ConsEner       EmisCO2
## CrecPobl     3.361974e-01 -8.384169e+02     -1.137877
## MortInf      1.276296e+01 -4.442568e+04    -94.855000
## PorcMujeres  2.680592e-01  2.855207e+02     -2.150132
## PNB95       -5.806203e+04  1.415628e+08 250167.323509
## ProdElec    -3.180340e+04  6.801296e+07 139277.888640
## LinTelf     -9.940461e+01  3.426262e+05    638.570000
## ConsAgua    -6.743793e+01  2.092242e+05    486.932763
## PropBosq     2.625263e+00 -5.153439e+03    -12.897193
## PropDefor    1.817253e+00 -1.051522e+03     -2.632487
## ConsEner    -1.051522e+03  5.014395e+06  10286.159781
## EmisCO2     -2.632487e+00  1.028616e+04     27.268614
```

```
print("Matriz Correlación:")
```

```
## [1] "Matriz Correlación:"
```

```
matrizCorr
```

```
##                CrecPobl     MortInf PorcMujeres       PNB95    ProdElec
## CrecPobl     1.00000000  0.55067948 -0.55852711 -0.32212154 -0.29711119
## MortInf      0.55067948  1.00000000 -0.03280139 -0.31579250 -0.24053689
## PorcMujeres -0.55852711 -0.03280139  1.00000000  0.14338826  0.19085114
## PNB95       -0.32212154 -0.31579250  0.14338826  1.00000000  0.74476081
## ProdElec    -0.29711119 -0.24053689  0.19085114  0.74476081  1.00000000
## LinTelf     -0.56321228 -0.69558922  0.26167018  0.46539599  0.28664508
## ConsAgua    -0.06772953 -0.06975563 -0.31106243  0.08541500  0.18047653
## PropBosq    -0.15650281 -0.02278415  0.37096694  0.05525919  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     -0.17568860 -0.56520778 -0.04692837  0.21425123  0.19760017
##                LinTelf    ConsAgua    PropBosq   PropDefor    ConsEner
## CrecPobl    -0.56321228 -0.06772953 -0.15650281  0.20107881 -0.30187731
## MortInf     -0.69558922 -0.06975563 -0.02278415  0.29459348 -0.61731132
## PorcMujeres  0.26167018 -0.31106243  0.37096694  0.02266339  0.01453216
## PNB95        0.46539599  0.08541500  0.05525919 -0.19262327  0.28272492
## ProdElec     0.28664508  0.18047653  0.02603078 -0.17478434  0.22501894
## LinTelf      1.00000000  0.10593934  0.06337138 -0.37623801  0.78068385
## ConsAgua     0.10593934  1.00000000 -0.19299225 -0.08705811  0.16259804
## PropBosq     0.06337138 -0.19299225  1.00000000  0.09725032 -0.11492480
## PropDefor   -0.37623801 -0.08705811  0.09725032  1.00000000 -0.34833836
## ConsEner     0.78068385  0.16259804 -0.11492480 -0.34833836  1.00000000
## EmisCO2      0.62393719  0.16227447 -0.12333592 -0.37396154  0.87965517
##                EmisCO2
## CrecPobl    -0.17568860
## MortInf     -0.56520778
## PorcMujeres -0.04692837
## PNB95        0.21425123
## ProdElec     0.19760017
## LinTelf      0.62393719
## ConsAgua     0.16227447
```

```
## PropBosq    -0.12333592
## PropDefor   -0.37396154
## ConsEner     0.87965517
## EmisCO2      1.00000000
```

2. Calcule los valores y vectores propios de cada matriz.La función en R es: eigen().

```
# eigen covarianza
print("Eigen Covarianza: ")

## [1] "Eigen Covarianza: "

eigenCov <- eigen(matrizCov)
eigenCov

## eigen() decomposition
## $values
##  [1] 6.163576e+10 6.581612e+09 4.636256e+06 3.107232e+05 1.216015e+04
##  [6] 5.137767e+02 3.627885e+02 4.542082e+01 5.800868e+00 1.438020e+00
## [11] 4.768083e-01
##
## $vectors
##                [,1]          [,2]          [,3]          [,4]
[,5]
##  [1,] -1.658168e-06  4.706785e-07  0.0001263736 -1.928408e-05 -
0.0055373971
##  [2,] -4.048139e-05 -1.774254e-05  0.0082253821 -2.493257e-03 -
0.0944030203
##  [3,]  5.739096e-06 -1.084543e-05  0.0001318149  5.538307e-03
0.0314036410
##  [4,]  8.880376e-01  4.597632e-01  0.0026022071 -3.893588e-04 -
0.0003327409
##  [5,]  4.597636e-01 -8.880405e-01  0.0005694896  1.096305e-03
0.0002207819
##  [6,]  3.504341e-04  4.016179e-04 -0.0619424889  7.641174e-03
0.9921404486
##  [7,]  2.625508e-04 -1.122118e-03 -0.0401453227 -9.991411e-01
0.0057795144
##  [8,]  4.089564e-06  7.790843e-06  0.0012719918  6.435797e-03
0.0419331615
##  [9,] -1.073825e-06  2.350808e-07  0.0001916177  4.043796e-05 -
0.0018090751
## [10,]  2.547156e-03  7.126782e-04 -0.9972315499  3.973568e-02 -
0.0625729475
## [11,]  4.643724e-06 -1.315731e-06 -0.0020679047 -5.626049e-05 -
0.0042367120
##                [,6]          [,7]          [,8]          [,9]
[,10]
##  [1,]  1.243456e-02  5.359089e-03 -8.390810e-02 -6.778358e-02 -1.158091e-
```

```
01
##  [2,]  9.917515e-01  2.258020e-02 -7.891128e-02 -1.637836e-02  4.264872e-
04
##  [3,]  8.552992e-02 -1.136481e-01  9.856498e-01 -1.468464e-02  8.241465e-
03
##  [4,] -8.621005e-06 -7.566477e-06  1.217248e-05 -3.971469e-07  4.274451e-
07
##  [5,]  1.955408e-05  1.544658e-05 -2.558998e-05  1.059471e-06 -1.353881e-
06
##  [6,]  9.109622e-02  4.748682e-02 -3.416812e-02 -5.379549e-03 -3.409423e-
03
##  [7,] -1.087229e-03 -6.863294e-03  4.698731e-03  7.965261e-05  3.621425e-
05
##  [8,]  1.721948e-02 -9.920538e-01 -1.169638e-01  1.416566e-03  5.891758e-
03
##  [9,]  1.758667e-03 -7.455427e-03  1.811443e-02  1.283039e-01 -9.859317e-
01
## [10,]  2.639673e-03 -3.764707e-03  1.267052e-03  2.262931e-03  2.672618e-
04
## [11,] -1.877994e-02 -1.709137e-03 -5.204823e-03 -9.891529e-01 -1.200519e-
01
##               [,11]
##  [1,]  9.872887e-01
##  [2,] -2.092491e-02
##  [3,]  8.344324e-02
##  [4,]  2.723996e-07
##  [5,] -2.086857e-07
##  [6,]  4.944397e-04
##  [7,]  4.780416e-04
##  [8,] -3.748976e-03
##  [9,] -1.052934e-01
## [10,]  5.906241e-05
## [11,] -8.221371e-02
```

```
# Valores y Vectores Propios Covarianza
valPropCov <- eigenCov$values
vectPropCov <- eigenCov$vectors

# eigen correlación
print("Eigen Correlación: ")
```

```
## [1] "Eigen Correlación: "
```

```
eigenCorr <- eigen(matrizCorr)
eigenCorr
```

```
## eigen() decomposition
## $values
##  [1] 4.02987902 1.92999195 1.37041115 0.86451597 0.79414057 0.72919997
##  [7] 0.57130511 0.32680096 0.16806846 0.14632819 0.06935866
##
```

```
## $vectors
##               [,1]        [,2]        [,3]        [,4]        [,5]
[,6]
##  [1,] -0.314119414  0.34835747 -0.07352541 -0.44028717 -0.32972147 -
0.18392437
##  [2,] -0.392395442 -0.04136238 -0.17759254 -0.13398483  0.08340489 -
0.08656390
##  [3,]  0.116546319 -0.58283641  0.16686305  0.05865031  0.18654100
0.16835650
##  [4,]  0.295393771 -0.17690839 -0.53343025 -0.26248209 -0.14110658
0.04653378
##  [5,]  0.258964724 -0.17356372 -0.61438847 -0.17389644 -0.07521971
0.02821905
##  [6,]  0.446082934 -0.02719077  0.15177250  0.04959796 -0.05416498
0.02442175
##  [7,]  0.092410503  0.32060987 -0.37024258  0.73603097  0.02671021 -
0.30940890
##  [8,]  0.005692925 -0.45742697  0.16480339  0.04024882 -0.41531702 -
0.75356463
##  [9,] -0.243652293 -0.15408201 -0.02961449  0.33650345 -0.73261463
0.50894232
## [10,]  0.415029554  0.23286257  0.20608749 -0.06730166 -0.23100421
0.05806466
## [11,]  0.374531032  0.29168698  0.20631751 -0.14843513 -0.24028756 -
0.02809233
##                [,7]        [,8]        [,9]       [,10]       [,11]
##  [1,]  0.1628974320 -0.09481963  0.52181220  0.34674573 -0.10062784
##  [2,]  0.6398040762 -0.32307802 -0.29031618 -0.38959240  0.17487096
##  [3,]  0.5310867107  0.05209889  0.23599758  0.42854658 -0.16786800
##  [4,] -0.1490207046 -0.44913216 -0.36995675  0.34911534 -0.15247432
##  [5,]  0.1082745817  0.50343911  0.30681318 -0.33770404  0.12366382
##  [6,] -0.0008501608 -0.56975094  0.44733110 -0.20997673  0.44992596
##  [7,]  0.2357666690 -0.05962470  0.08358225  0.20561803 -0.07067780
##  [8,] -0.0806036686  0.04275404 -0.07438520 -0.08671232 -0.01493710
##  [9,]  0.0112333588 -0.01607505 -0.01868615 -0.03209758  0.07259619
## [10,]  0.2711228006 -0.05023582 -0.04339752 -0.36147417 -0.67912543
## [11,]  0.3352822144  0.30978009 -0.37666244  0.28779437  0.46737561

#Valores y Vectores Propios Correlación
valPropCorr <- eigenCorr$values
vectPropCorr <- eigenCorr$vectors
```

3.1. Calcule la proporción de varianza explicada por cada componente. Se sugiere dividir cada lambda entre la varianza total (las lambdas están en eigen(S)[1]). La varianza total es la suma de las varianzas de la diagonal de S. Una forma es sum(diag(S)). La varianza total de los componentes es la suma de los valores propios (es decir, la suma de la varianza de cada componente), sin embargo, si sumas la diagonal de S (es decir, la varianza de cada x), te da el mismo valor (¡comprúebalo!). Recuerda que las combinaciones lineales buscan reproducir la varianza de X.

```
totVarianzaCov <- sum(diag(matrizCov))
propVarExpCov <- valPropCov / totVarianzaCov

print("Proporción de Varianza Explicada: ")

## [1] "Proporción de Varianza Explicada: "

propVarExpCov

##  [1] 9.034543e-01 9.647298e-02 6.795804e-05 4.554567e-06 1.782429e-07
##  [6] 7.530917e-09 5.317738e-09 6.657763e-10 8.502887e-11 2.107843e-11
## [11] 6.989035e-12
```

4.1. Acumule los resultados anteriores. (cumsum() puede servirle).

```
propVarAcumCov <- cumsum(propVarExpCov)
propVarAcumCov

##  [1] 0.9034543 0.9999273 0.9999953 0.9999998 1.0000000 1.0000000 1.0000000
##  [8] 1.0000000 1.0000000 1.0000000 1.0000000
```

5.1. Según los resultados anteriores, ¿qué componentes son los más importantes? ¿qué variables son las que más contribuyen a la primera y segunda componentes principales? ¿por qué lo dice? ¿influyen las unidades de las variables?

- Para covarianza los más importantes serían el primero y el segundo componente. La sexta variable es la que está afectando al primer componente principal y en la segunda variable principal, la que está afectando es la tercera variable. Esto se puede comprobar con el peso de cada variable en el componente principal que se visualizó previamente. Por lo anterior también llegamos a la conclusión de que las unidades de las variables influyen en el resultado.

3.2. Calcule la proporción de varianza explicada por cada componente. Se sugiere dividir cada lambda entre la varianza total (las lambdas están en eigen(S)[1]). La varianza total es la suma de las varianzas de la diagonal de S. Una forma es sum(diag(S)). La varianza total de los componentes es la suma de los valores propios (es decir, la suma de la varianza de cada componente), sin embargo, si sumas la diagonal de S (es decir, la varianza de cada x), te da el mismo valor (¡comprúebalo!). Recuerda que las combinaciones lineales buscan reproducir la varianza de X.

```
totVarianzaCorr <- sum(diag(matrizCorr))
propVarExpCorr <- valPropCorr / totVarianzaCorr

print("Proporción de Varianza Explicada: ")

## [1] "Proporción de Varianza Explicada: "

propVarExpCorr

##  [1] 0.366352638 0.175453813 0.124582832 0.078592361 0.072194597
0.066290906
##  [7] 0.051936828 0.029709178 0.015278951 0.013302563 0.006305332
```

4.2. Acumule los resultados anteriores. (cumsum() puede servirle).

```
propVarAcumCorr <- cumsum(propVarExpCorr)
propVarAcumCorr

##  [1] 0.3663526 0.5418065 0.6663893 0.7449816 0.8171762 0.8834671 0.9354040
##  [8] 0.9651132 0.9803921 0.9936947 1.0000000
```

5.2 Según los resultados anteriores, ¿qué componentes son los más importantes? ¿qué variables son las que más contribuyen a la primera y segunda componentes principales? ¿por qué lo dice? ¿influyen las unidades de las variables?

- En la correlación podemos observar que solo los primeros dos componentes son los necesarios. Si comparamos ambos resultados, podemos llegar a la conclusión de que con solo los dos componentes principales son necesarios.

7. Compare los resultados de los incisos 6 y 7. ¿qué concluye?

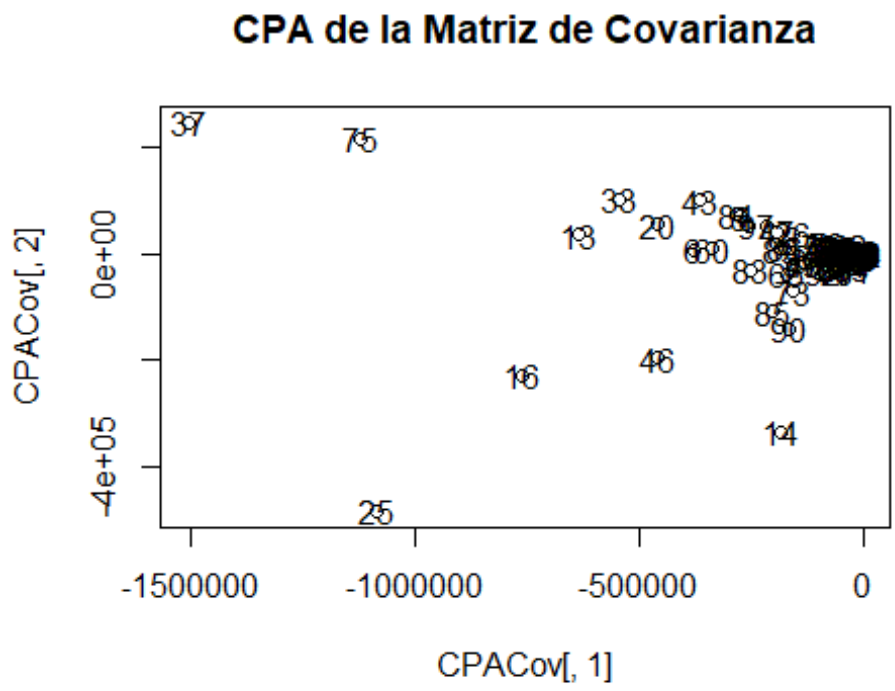
#Parte 2

```r
#install.packages("factoextra")
#install.packages("ggplot2")

# Cargamos librerias
library(stats)
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa
```

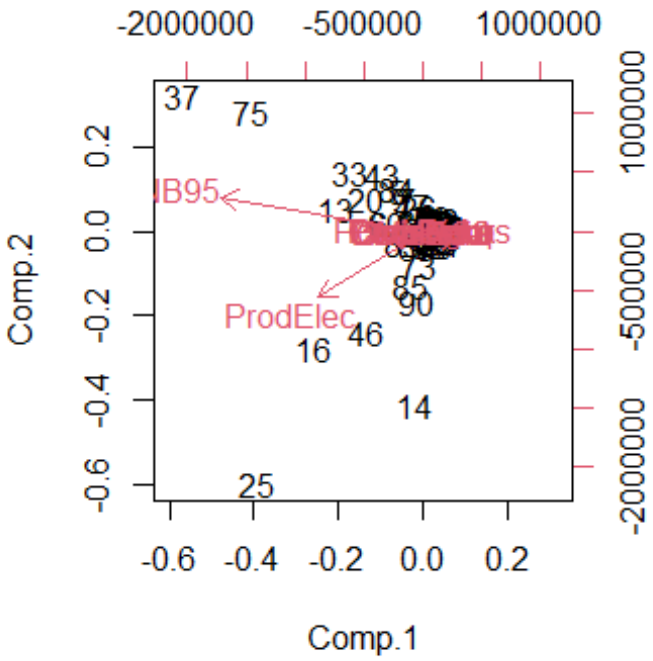
```r
library(ggplot2)

# Realizar el PCA para la matriz de covarianzas
CPCov <- princomp(data, cor = FALSE)
CPACov <- as.matrix(data) %*% CPCov$loadings

# Gráficamos las dos primeras componentes para covarianza
plot(CPACov[, 1], CPACov[, 2], type = "p", main = "CPA de la Matriz de
Covarianza")
text(CPACov[, 1], CPACov[, 2], 1:nrow(CPACov))
```



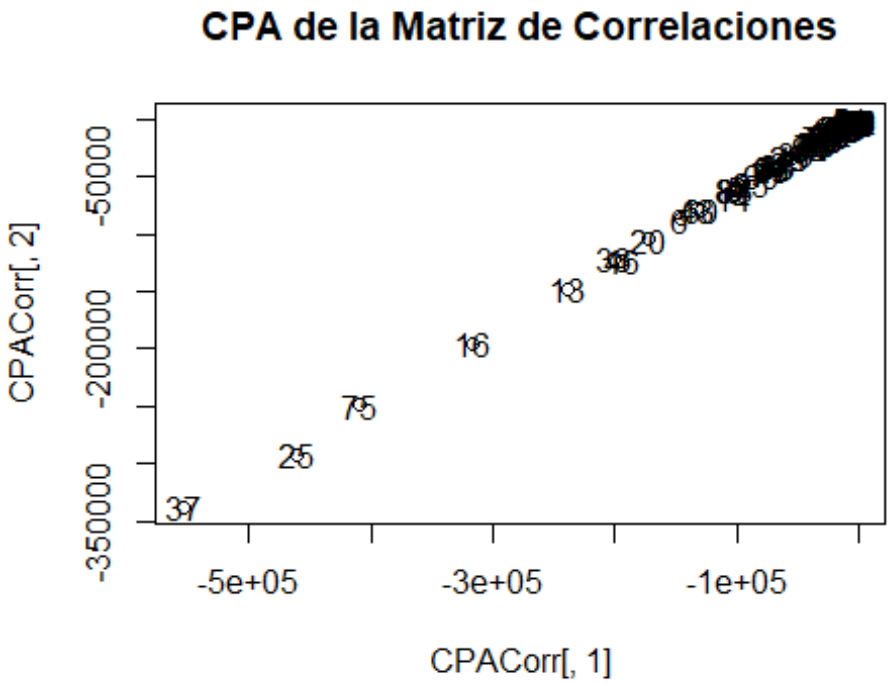

```r
biplot(CPCov)
```

```
## 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
```

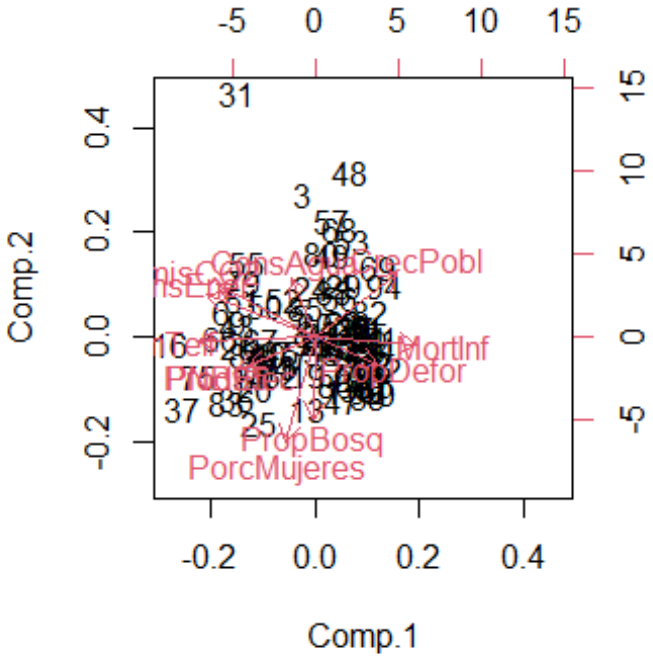
```
## Warning in arrows(0, 0, y[, 1L] * 0.8, y[, 2L] * 0.8, col = col[2L],
length =
## arrow.len): zero-length arrow is of indeterminate angle and so skipped

## 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
```




```r
# Realizar el PCA para la matriz de correlación
CPCorr <- princomp(data, cor = TRUE)
CPACorr <- as.matrix(data) %*% CPCorr$loadings

# Gráfica de las dos primeras componentes para correlación
plot(CPACorr[, 1], CPACorr[, 2], type = "p", main = "CPA de la Matriz de
Correlaciones")
text(CPACorr[, 1], CPACorr[, 2], 1:nrow(CPACorr))
```

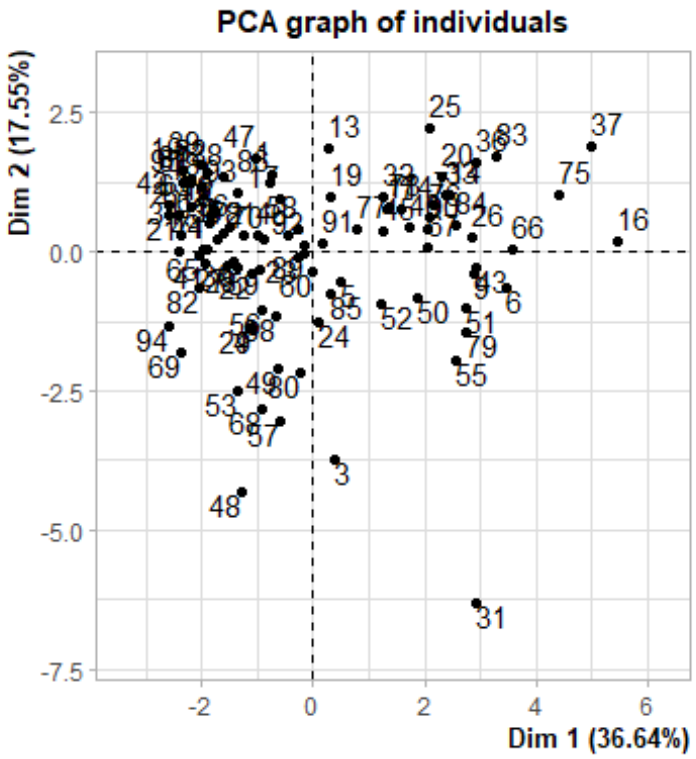


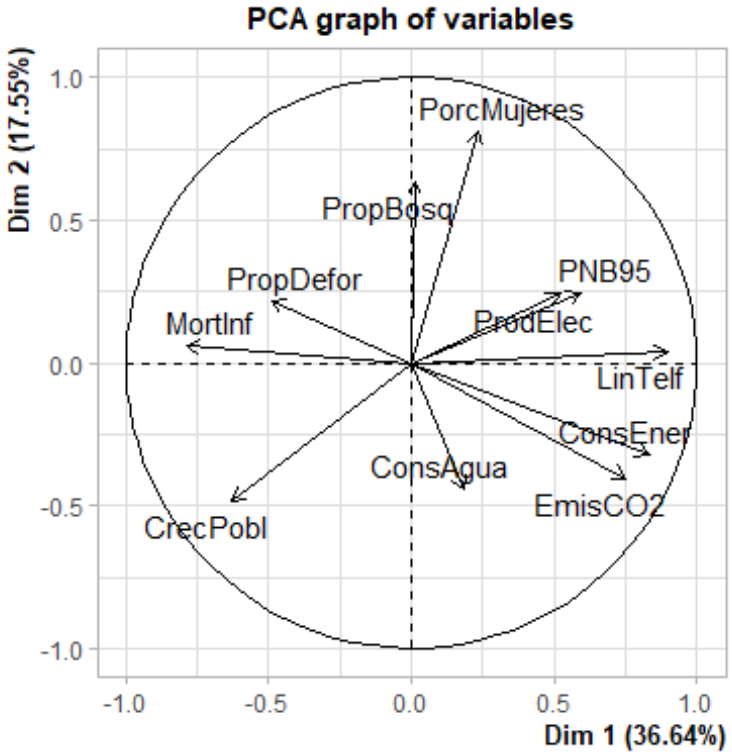

```
biplot(CPCorr)
```



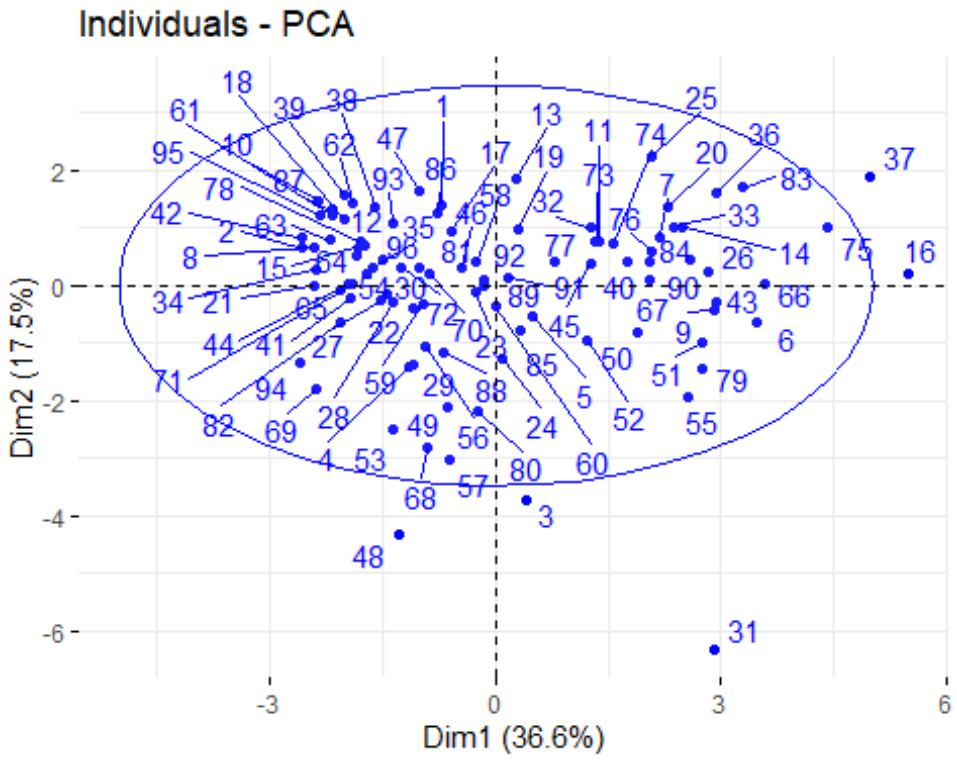
Parte 3

```r
# Cargamos librerias
library(FactoMineR)
library(factoextra)
library(ggplot2)

PCA3 <- PCA(data)
```

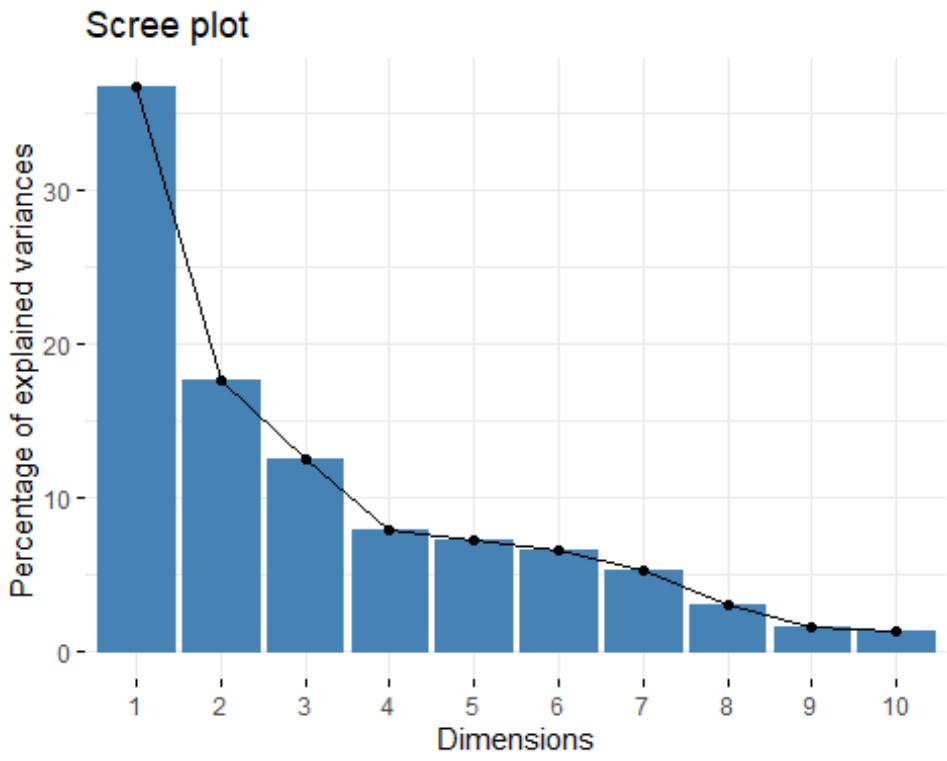




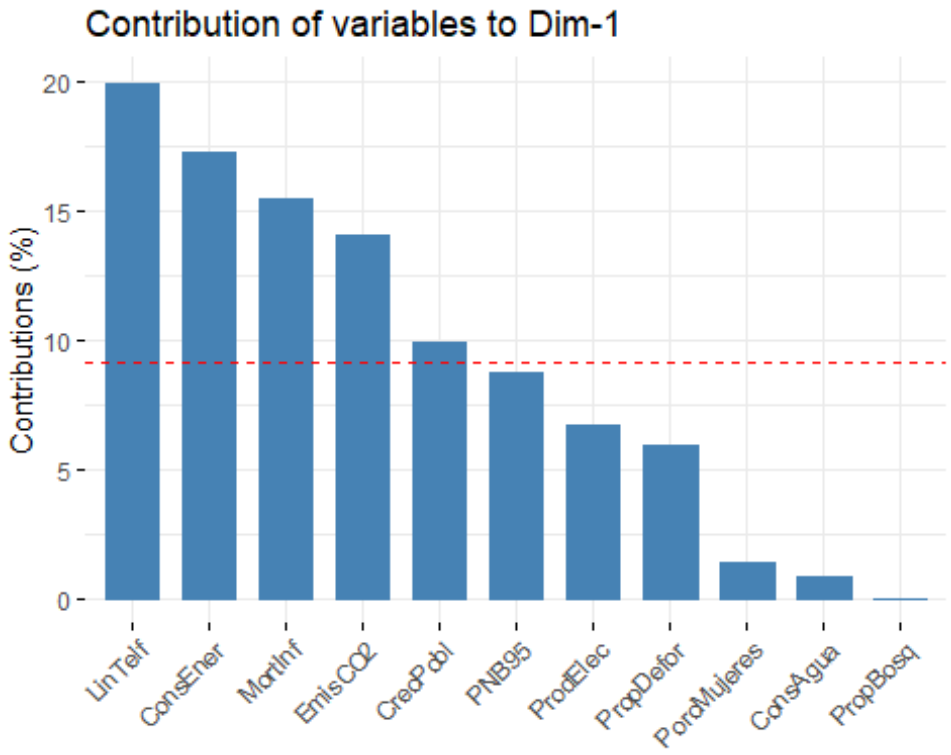

```
fviz_pca_ind(PCA3, col.ind = "blue", addEllipses = TRUE, repel = TRUE)
```




```
fviz_screeplot(PCA3)
```




```
fviz_contrib(PCA3, choice = c("var"))
```


Contribution of variables to Dim-1
Contributions (%)
20
15
10
5
0
UnTelf
ConsEner
MortInf
EmisCO2
CreaPobl
PNB95
ProdElec
PropDefor
PoroMujeres
ConsAgua
PropBosq