Actividad 1.10

Cesar Vazquez

2022-11-17

```
M = read.csv("datos_dentrifico.csv")
names(M)
```

```
## [1] "V1" "V2" "V3" "V4" "V5" "V6"
```

- Se trata de 30 observaciones con 6 variables.
- Se trata de variables categóricas ordinales.
- Suponiendo equisdistancia entre las opciones, nos arriesgamos a considerar las variables numéricas discretas.

Descripción

```
apply(M, 2, summary) # 2 = operar por columna, función
```

```
##               V1  V2  V3  V4  V5       V6
## Min.    1.000000 2.0 1.0 2.0 1.0 2.000000
## 1st Qu. 2.000000 3.0 2.0 3.0 2.0 3.000000
## Median  4.000000 4.0 4.0 4.0 3.5 4.000000
## Mean    3.933333 3.9 4.1 4.1 3.5 4.166667
## 3rd Qu. 6.000000 5.0 6.0 5.0 5.0 4.750000
## Max.    7.000000 7.0 7.0 7.0 7.0 7.000000
```

```
apply(M, 2, sd) # sd = desv estándar de muestra s
```

```
##       V1       V2       V3       V4       V5       V6
## 1.981524 1.373392 2.056948 1.373392 1.907336 1.391683
```

```
n = nrow(M)
cat("Num. de observaciones:", n)
```

```
## Num. de observaciones: 30
```

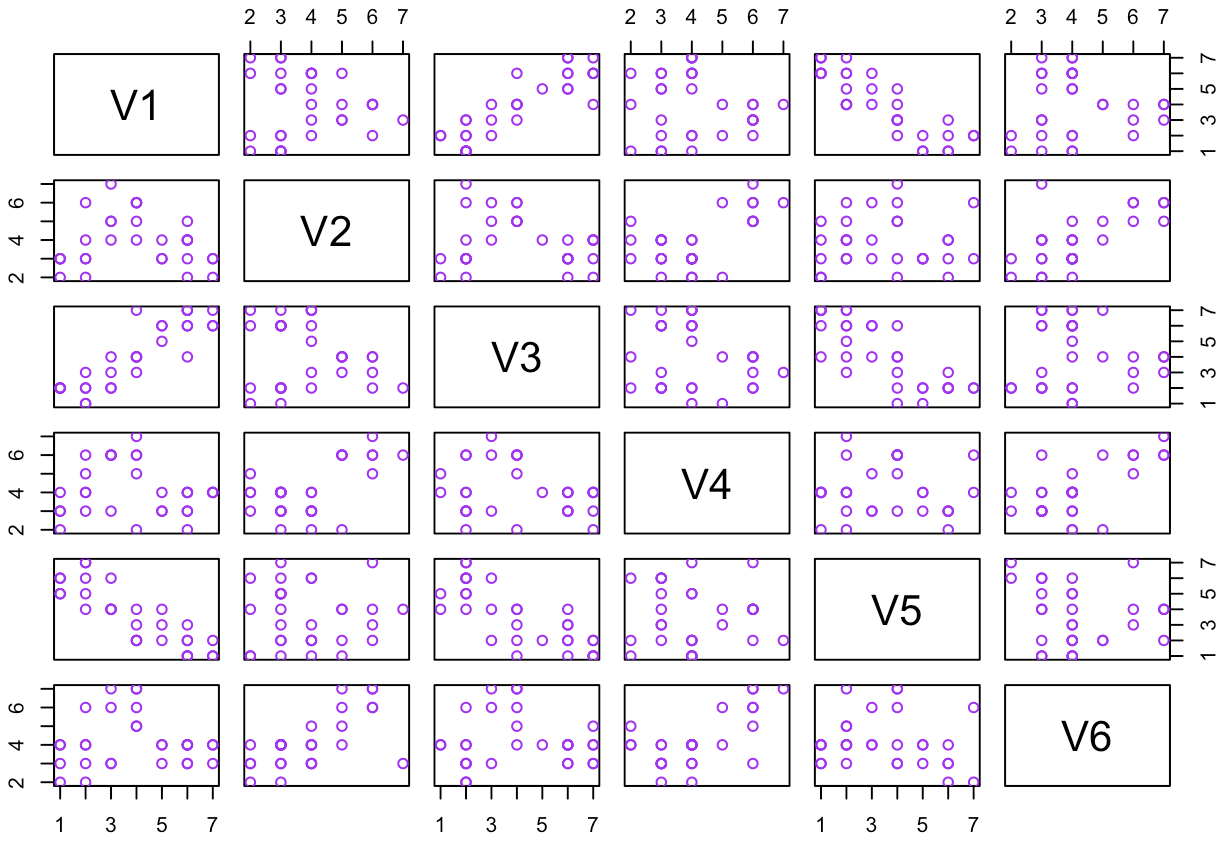
*Se puede decir que los datos podrian ser normales, ya que la media y la mediana son parecidos.

**Se observan posibles distribuciones semejantes y tal vez simétricas por la cercanía de la media y mediana, y similares valores de la desviación estándar.

** El tamaño de la muestra (núm. de obs.) es muy pequeño, es no eficiente.

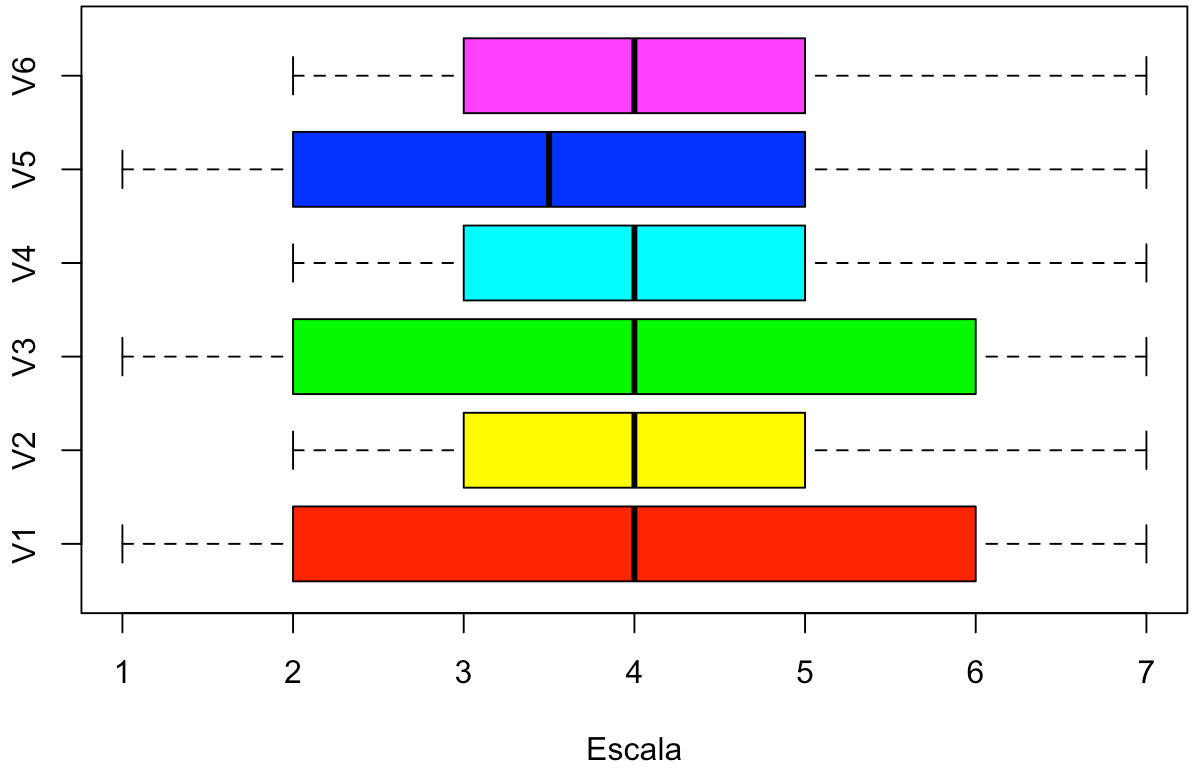
Descripción gráfica

```
plot(M, col = "purple")
```



Se notan algunas variables correlaciones como V1-V3, V2-V6,

```
boxplot(M, horizontal = T, col = rainbow(6), main = "Datos del dentrifico", xlab = "E
scala")
```



Se observa cierta simetría interna de las variables, excepto V5. Las medianas son similares en todos excepto en la V5.

Analisis correlacional (por pares)

```
round(cor(M),3)
```

```
##        V1     V2     V3     V4     V5     V6
## V1  1.000 -0.053  0.873 -0.086 -0.858  0.004
## V2 -0.053  1.000 -0.155  0.572  0.020  0.640
## V3  0.873 -0.155  1.000 -0.248 -0.778 -0.018
## V4 -0.086  0.572 -0.248  1.000 -0.007  0.640
## V5 -0.858  0.020 -0.778 -0.007  1.000 -0.136
## V6  0.004  0.640 -0.018  0.640 -0.136  1.000
```

V1 se observa una alta correlación con V3 y V5, mientras que V2 se observa V4 y V6.

$H_o : \rho_{i,j} = 0$ #rho significa correlación de población.

$H_1 : \rho_{i,j} \neq 0$
$\alpha = 0.05$
Regla de decisión: Si valor p < alfa = 0.05, se rechaza H_o

```
library(Hmisc)
correl = rcorr(as.matrix(M),type="spearman")
correl
```

```
##       V1    V2    V3    V4    V5    V6
## V1  1.00 -0.01  0.86 -0.04 -0.87  0.05
## V2 -0.01  1.00 -0.04  0.43  0.01  0.59
## V3  0.86 -0.04  1.00 -0.19 -0.79  0.06
## V4 -0.04  0.43 -0.19  1.00 -0.03  0.52
## V5 -0.87  0.01 -0.79 -0.03  1.00 -0.17
## V6  0.05  0.59  0.06  0.52 -0.17  1.00
##
## n= 30
##
##
## P
##    V1     V2     V3     V4     V5     V6
## V1        0.9453 0.0000 0.8413 0.0000 0.7802
## V2 0.9453        0.8195 0.0182 0.9597 0.0007
## V3 0.0000 0.8195        0.3052 0.0000 0.7451
## V4 0.8413 0.0182 0.3052        0.8567 0.0030
## V5 0.0000 0.9597 0.0000 0.8567        0.3627
## V6 0.7802 0.0007 0.7451 0.0030 0.3627
```

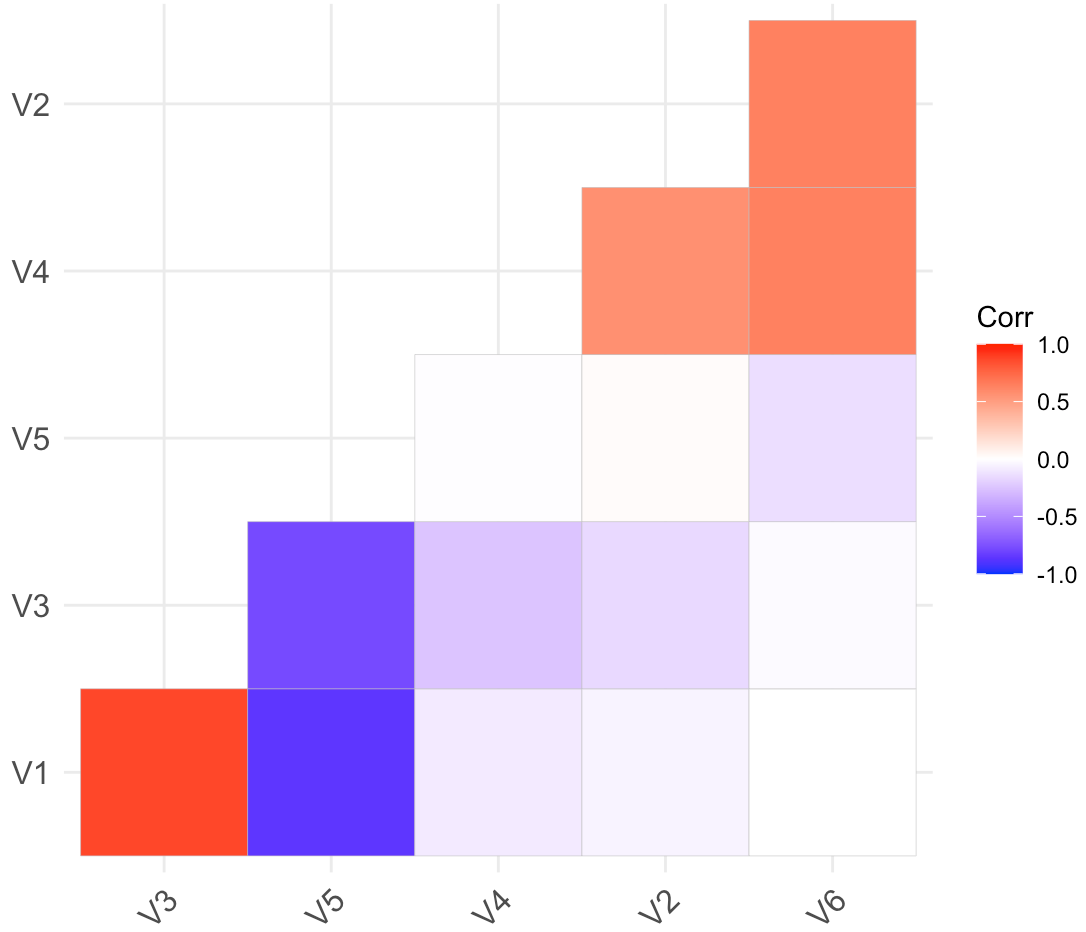
Hay correlación significativa ente:

V1, V2, V3

V2, V4, V6

Gráfica de correlaciones.

```
library(ggcorrplot)
ggcorrplot(cor(M), type = "lower", hc.order = T)
```



Se observan al menos 2 grupos de variables correlacionadas por pares.

Correlación conjunta.

H_o: Los datos provienen de una población de variables no correlacionadas en forma conjunta.

H_1: no H_o

```
library(performance)
check_sphericity_bartlett(M)
```

```
## # Test of Sphericity
##
## Bartlett's test of sphericity suggests that there is sufficient significant correl
ation in the data for factor analysis (Chisq(15) = 111.31, p < .001).
```

Hay suficiente significancia.

Prueba KMO:

```
library(psych)
KMO(cor(M))
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = cor(M))
## Overall MSA =  0.66
## MSA for each item =
##   V1   V2   V3   V4   V5   V6
## 0.62 0.70 0.68 0.64 0.77 0.56
```

Como MSA > 0.5, es aceptable un análisis de componentes principales o factorial.

Proximamente: - Mardia test - QQplot

```
vMedia = colMeans(M)
S = cov(M)
Dm = mahalanobis(M, vMedia, S)
gl = ncol(M)
# qchisq me das una area a la izquierda y te doy la x
for(i in c(0.25, 0.5, 0.75)){
  prop = sum(Dm < qchisq(i,gl))/length(Dm)
  cat("Observado:",prop, "Esperado: ", i*100, "%\n")
}
```

```
## Observado: 0.2333333 Esperado:  25 %
## Observado: 0.5666667 Esperado:  50 %
## Observado: 0.8333333 Esperado:  75 %
```

Comentario: No muy similares, pero son cercanos.

```
library(MVN)
mvn(data = M, mvnTest = "mardia")
```

```
## $multivariateNormality
##              Test        Statistic             p value Result
## 1 Mardia Skewness 90.7322011256891 0.00227863080959803     NO
## 2 Mardia Kurtosis 1.15981062814193   0.246125915955431    YES
## 3             MVN             <NA>                <NA>     NO
##
## $univariateNormality
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling    V1        0.7472    0.0460    NO
## 2 Anderson-Darling    V2        1.0898    0.0062    NO
## 3 Anderson-Darling    V3        1.2672    0.0022    NO
## 4 Anderson-Darling    V4        1.2608    0.0023    NO
## 5 Anderson-Darling    V5        0.8201    0.0301    NO
## 6 Anderson-Darling    V6        1.6222    0.0003    NO
##
## $Descriptives
##     n     Mean  Std.Dev Median Min Max 25th 75th      Skew   Kurtosis
## V1 30 3.933333 1.981524    4.0   1   7    2 6.00 0.0119199 -1.3833659
## V2 30 3.900000 1.373392    4.0   2   7    3 5.00 0.4817598 -0.8011453
## V3 30 4.100000 2.056948    4.0   1   7    2 6.00 0.1001951 -1.5528878
## V4 30 4.100000 1.373392    4.0   2   7    3 5.00 0.3674963 -0.9360611
## V5 30 3.500000 1.907336    3.5   1   7    2 5.00 0.2738243 -1.2296957
## V6 30 4.166667 1.391683    4.0   2   7    3 4.75 0.6774265 -0.4487887
```

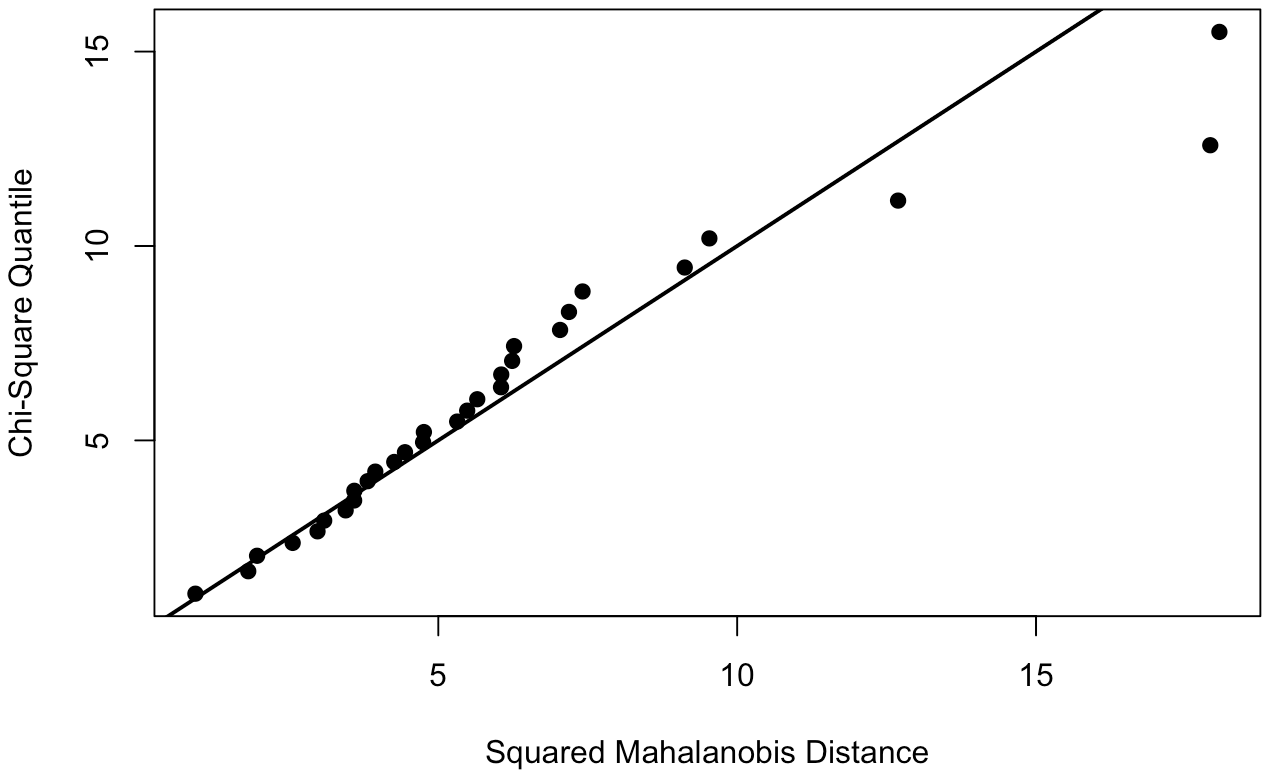
Según la prueba de normalidad de Marbia no pasa la prueba de sesgo, pero si la de kurtosis. Los datos no se deistribuyen normal.

```
mvn(data = M, mvnTest = "hz")
```

```
## $multivariateNormality
##            Test        HZ    p value MVN
## 1 Henze-Zirkler 0.9847967 0.01738351  NO
##
## $univariateNormality
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling    V1        0.7472    0.0460    NO
## 2 Anderson-Darling    V2        1.0898    0.0062    NO
## 3 Anderson-Darling    V3        1.2672    0.0022    NO
## 4 Anderson-Darling    V4        1.2608    0.0023    NO
## 5 Anderson-Darling    V5        0.8201    0.0301    NO
## 6 Anderson-Darling    V6        1.6222    0.0003    NO
##
## $Descriptives
##     n     Mean  Std.Dev Median Min Max 25th 75th      Skew   Kurtosis
## V1 30 3.933333 1.981524    4.0   1   7    2 6.00 0.0119199 -1.3833659
## V2 30 3.900000 1.373392    4.0   2   7    3 5.00 0.4817598 -0.8011453
## V3 30 4.100000 2.056948    4.0   1   7    2 6.00 0.1001951 -1.5528878
## V4 30 4.100000 1.373392    4.0   2   7    3 5.00 0.3674963 -0.9360611
## V5 30 3.500000 1.907336    3.5   1   7    2 5.00 0.2738243 -1.2296957
## V6 30 4.166667 1.391683    4.0   2   7    3 4.75 0.6774265 -0.4487887
```

QQPLOT

```
test = mvn(data = M, multivariatePlot = "qq")
```



Comentario: se observan que las distancias de Mahalanobis mayores provocan sesgo positivo.

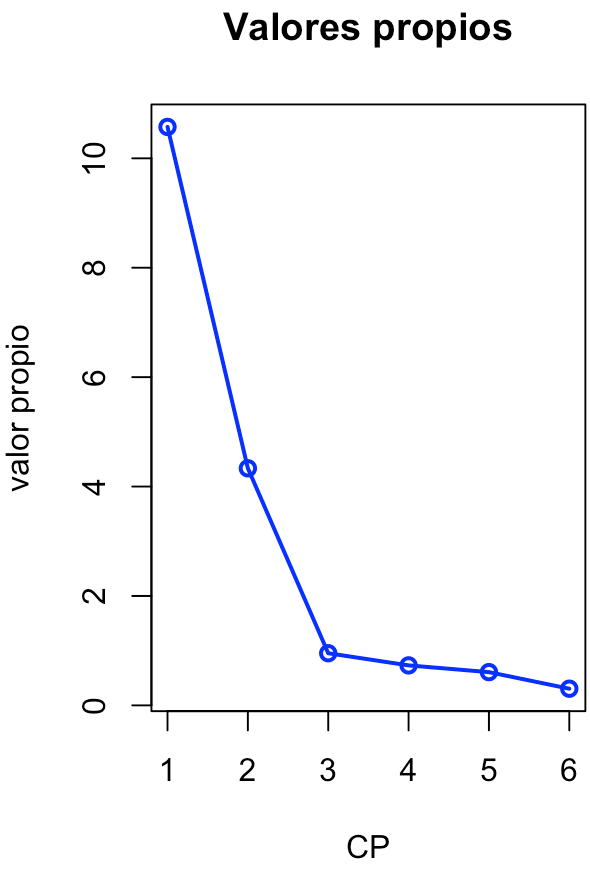
Análisis de componentes principales.

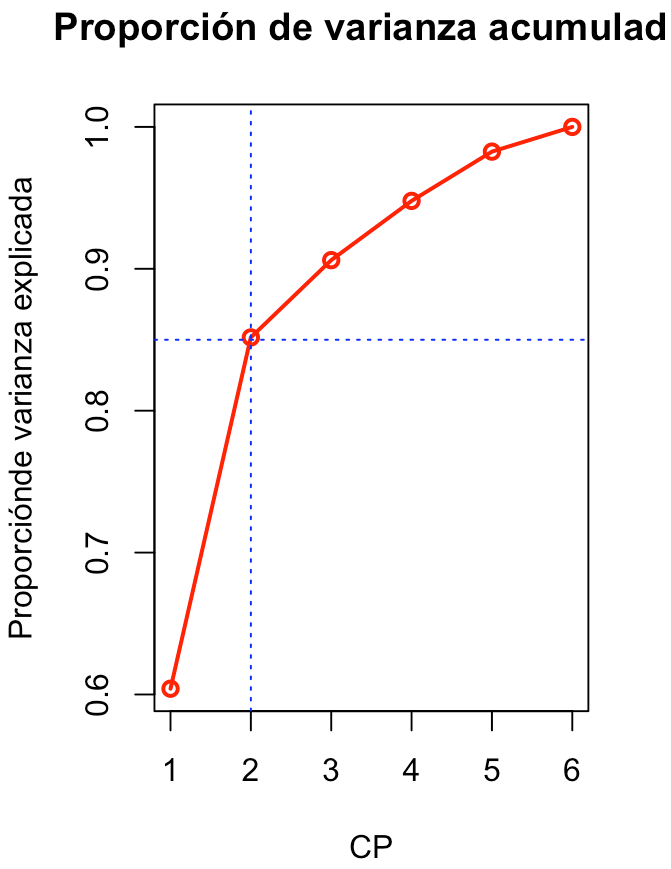
Nota. No demanda normalidad

```
S = cov(M)
lye = eigen(S)
VarTotal = sum(diag(S))
#sum(lye$values)
Prop_var_explicada = cumsum(lye$values/VarTotal)
Prop_var_explicada
```

```
## [1] 0.6040845 0.8516494 0.9061036 0.9478669 0.9825809 1.0000000
```

```
par(mfrow = c(1,2))
plot(1:6,lye$values,type = "o", lwd = 2, col = "blue", xlab = "CP", ylab = "valor pro
pio",main =  "Valores propios")
plot(1:6,Prop_var_explicada,type = "o", lwd = 2, col = "red", xlab = "CP", ylab = "Pr
oporciónde varianza explicada",  main = "Proporción de varianza acumulada")
abline(v = 2, lty = 3, col = "blue")
abline(h = 0.85, lty = 3, col = "blue")
```





Combinaciones lineales de los primeros componentes

```
CP12 = data.frame(lye$vectors[, c(1,2)])
CP12 = round(CP12,3)
colnames(CP12) = c("CP1", "CP2")
rownames(CP12) = c("V1", "V2", "V3", "V4", "V5", "V6")
```

Combinación lineal de CP1 y CP2: CP1 = 0.587V1 - 0.051V2 + 0.599V3 - 0.069V4 - 0.538V5 + 0.003V6 CP2 = -0.049V1 - 0.551V2 + 0.080V3 - 0.559V4 + 0.157V5 - 0.592V6

```
puntuaciones =  t(t(lye$vectors)%*%t(M))
head(puntuaciones,3)
```

```
##           [,1]      [,2]     [,3]       [,4]      [,5]      [,6]
## [1,]  6.205331 -5.803181 3.599659 -0.1169219 -6.235782 2.4414689
## [2,] -1.325084 -5.358310 3.511649  0.8710601 -5.128385 1.0686060
## [3,]  6.804591 -4.687856 2.944036  0.4831594 -6.093160 0.8331594
```

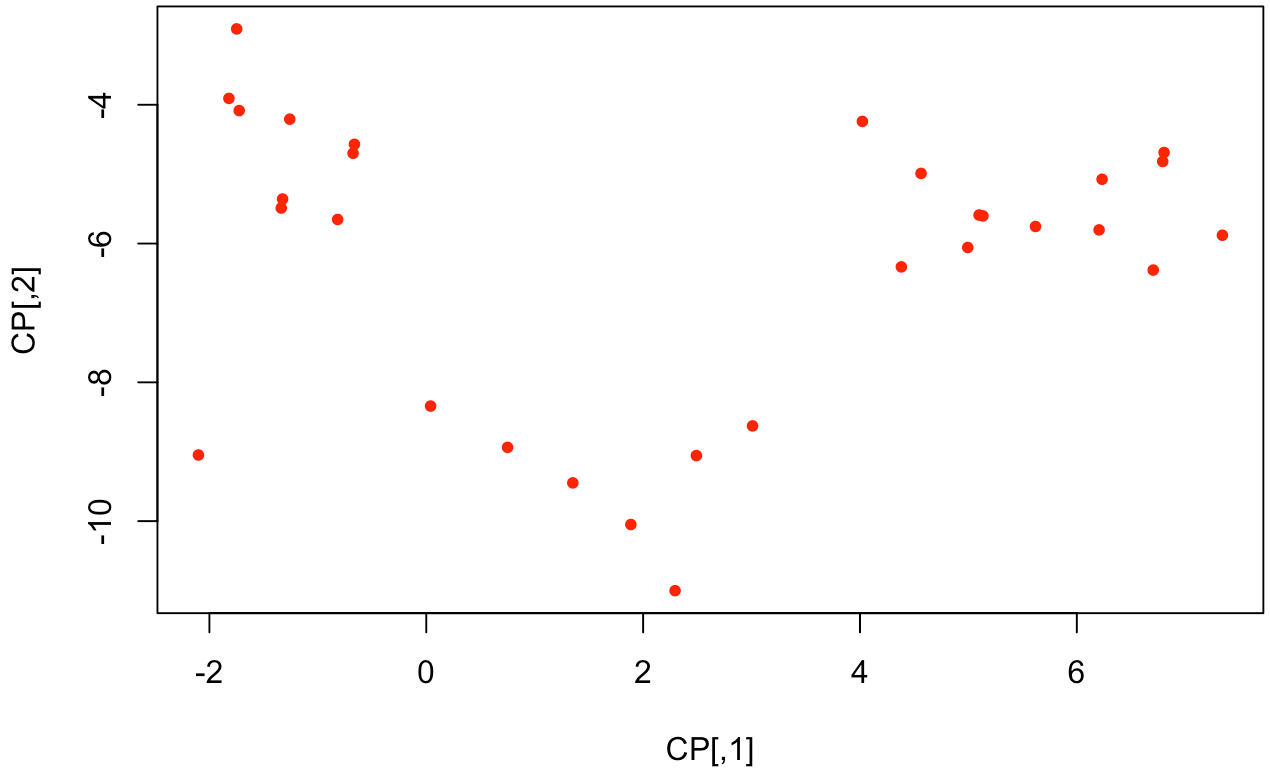
```
# Se seleccionan los componentes que afectan más a mis datos, ya que cubren el 85% de la varianza total.
CP = puntuaciones[,1:2]
head(CP, 8) # de 30
```

```
##           [,1]      [,2]
## [1,]  6.205331 -5.803181
## [2,] -1.325084 -5.358310
## [3,]  6.804591 -4.687856
## [4,]  3.009071 -8.627991
## [5,] -1.748098 -2.906781
## [6,]  5.618707 -5.754116
## [7,]  4.022299 -4.239424
## [8,]  6.704512 -6.381889
```

Se tratan de dos componentes ortogonales (por definición los componentes principales son).

Gráfica de CP1 y CP2

```
# No se pudo poner los vectores.
plot(CP, pch = 20, col = rainbow(1))
a = 1000
arrows(0, 0, a * lye$vectors[1,1], a*lye$vectors[1,2], code = 2)
```



Con librerias

prcomp: versión básica princomp: completa PCA: avanzada(está en FactoMineR)

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve
3WBa
```

```
cp = princomp(M, center = T, cor = T, scores = T); cp
```

```
## Warning: In princomp.default(M, center = T, cor = T, scores = T) :
##  extra argument 'center' will be disregarded
```

```
## Call:
## princomp(x = M, cor = T, scores = T, center = T)
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6
## 1.6526307 1.4893352 0.6645283 0.5841726 0.4273502 0.2919051
##
##  6  variables and  30 observations.
```

```
biplot(x = cp, scale= 0, cex=0.6, col = c("blue","brown3"), arrow.len = 0.1, expand =
0.9)
```

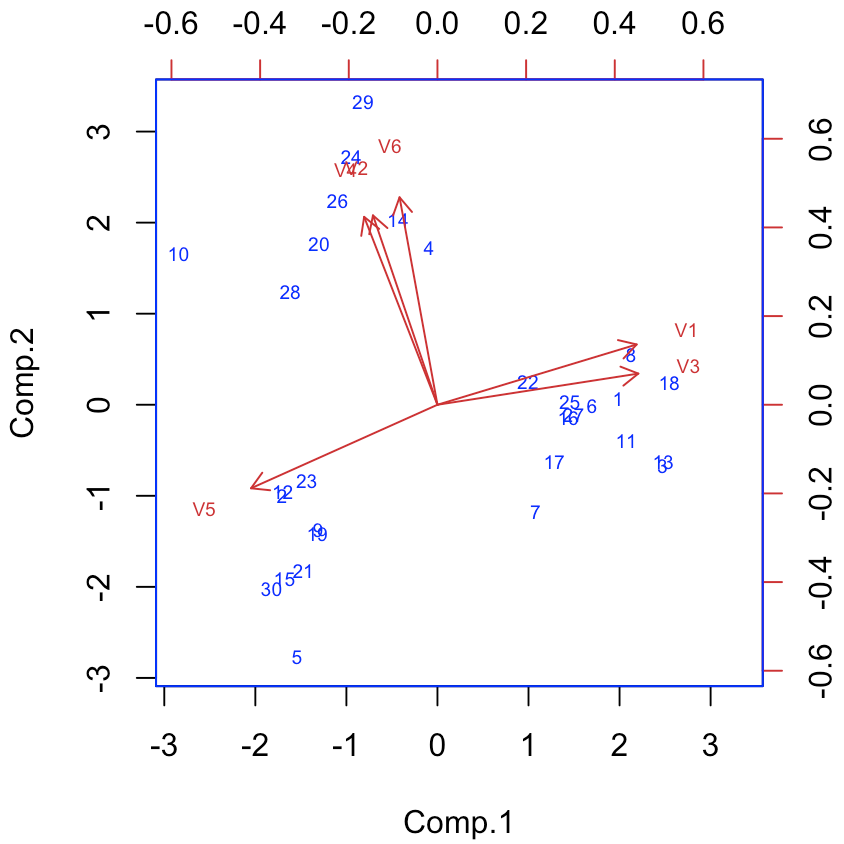
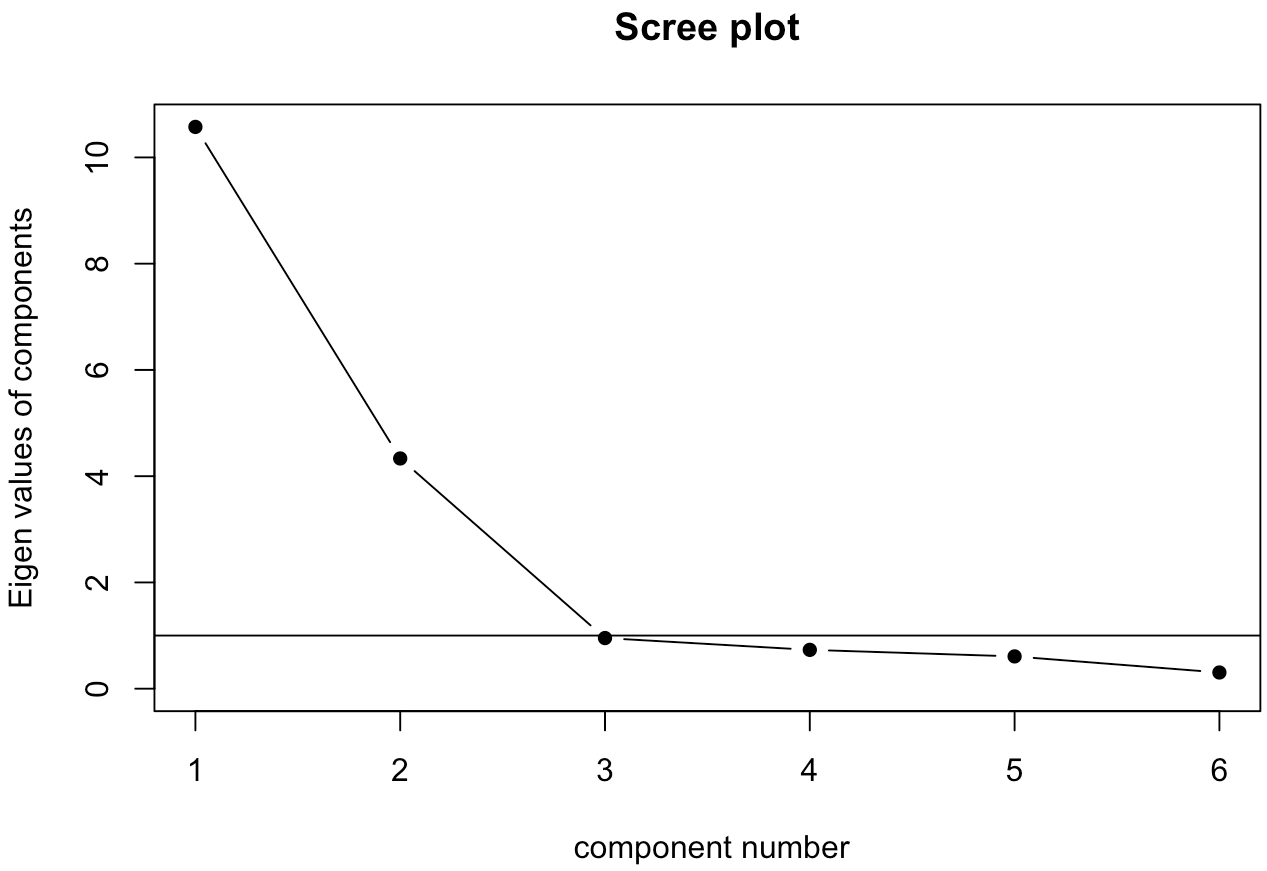


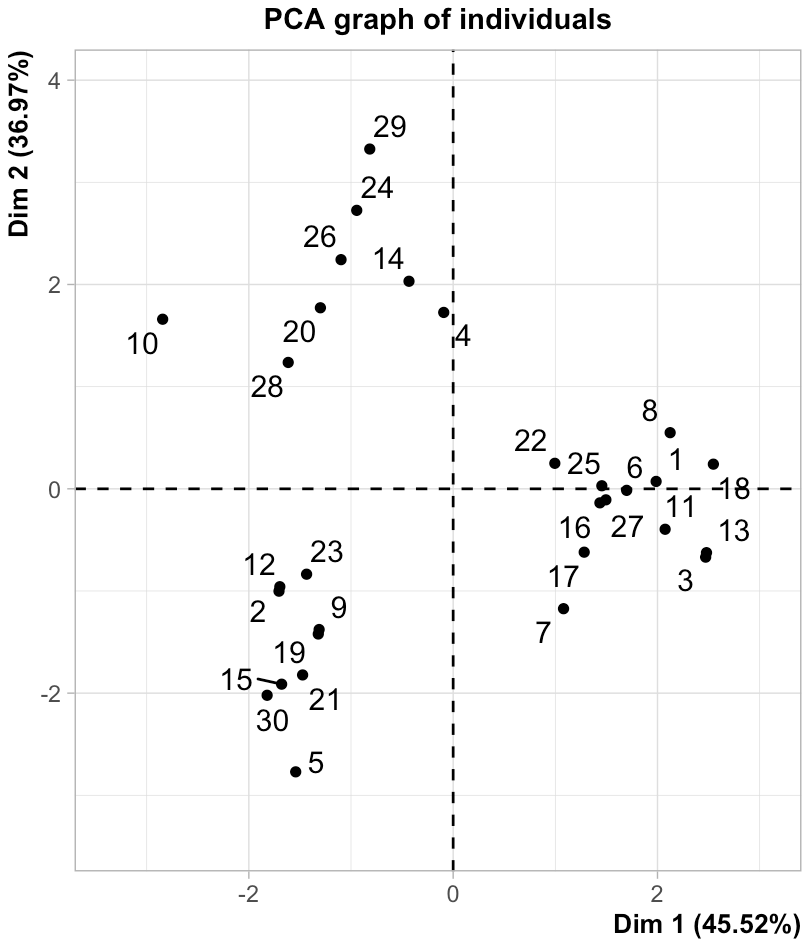
Grafico de Cattel

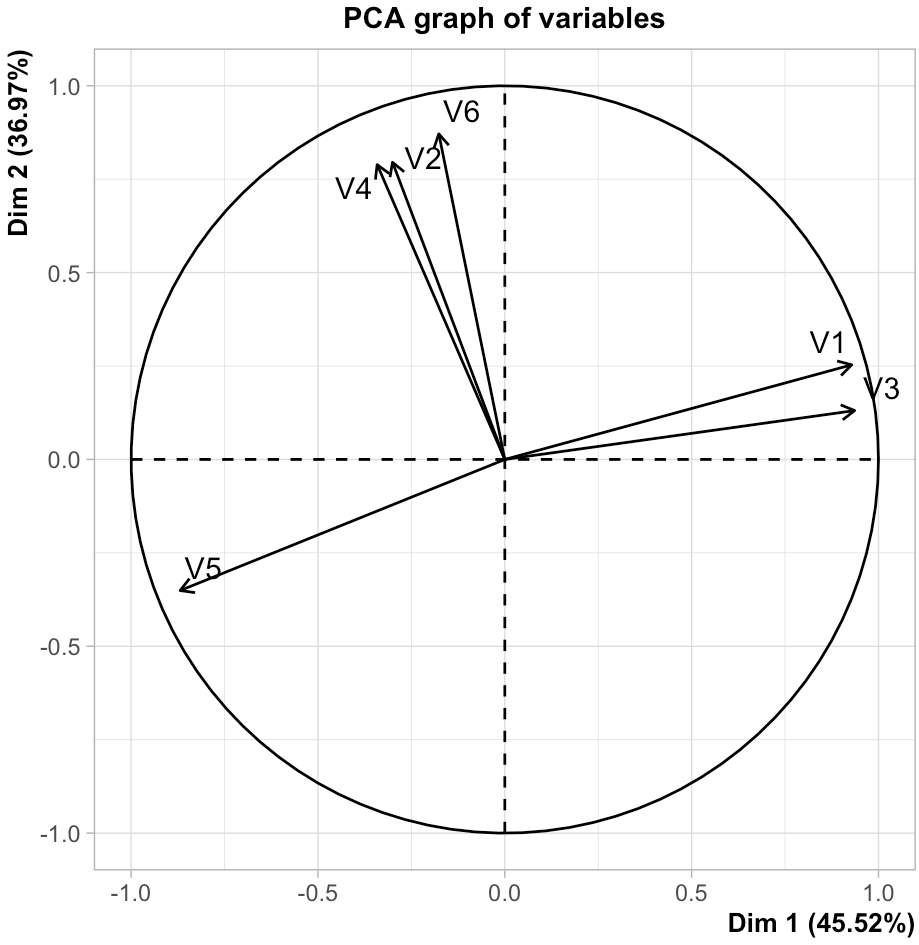
```
library(psych)
scree(cov(M), factors= F)
```



PCA

```
library(FactoMineR)
ACP = PCA(M, graph = T)
```


PCA graph of individuals
Dim 2 (36.97%)
Dim 1 (45.52%)
4
2
0
-2
-2
0
2
29
24
26
14
10
20
4
28
8
22
25
6
1
18
11
16
27
13
12
23
17
3
2
9
7
19
15
21
30
5


PCA graph of variables
Dim 2 (36.97%)
Dim 1 (45.52%)
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0
V6
V2
V4
V1
V3
V5

Análisis factorial

Con librerias

Factor_analysis —> Simple, está en library(parameters) rotation: none, “varimax”, quartimax, proma, oblimin, simplimax, cluster, … n = número de factores fa —> avanzado está en library(psych) MFA —> avanzado está en library(FactoMineR)