

# Análisis Factorial: Garcia

Oscar Eli Bonilla Morales

2022-05-23

## Análisis factorial: Ejemplo Garcia

Comenzamos con la carga de las paqueterías necesarias para nuestro análisis

```
install.packages("psych")
install.packages("polycor")
install.packages("ggcorrplot")
library(psych)
library(polycor)
library(ggcorrplot)
```

## Extracción de datos

Estos datos contienen 129 observaciones y 6 variables, entre estas se encuentran “protest”, “sexism” “anger”, entre otras.

```
x <- Garcia
head(x)
```

```
##   protest sexism anger liking respappr prot2
## 1      2   4.87    2   4.83    4.25    1
## 2      0   4.25    1   4.50    5.75    0
## 3      2   5.00    3   5.50    4.75    1
## 4      2   5.50    1   5.66    7.00    1
## 5      2   5.62    1   6.16    6.75    1
## 6      1   5.75    1   6.00    5.50    1
```

## Exploración de matriz

```
dim(x)
```

```
## [1] 129  6
```

## Tipos de variables

```
str(x)
```

```
## 'data.frame':  129 obs. of  6 variables:
## $ protest : num  2 0 2 2 2 1 2 0 0 0 ...
## ..- attr(*, "value.labels")= Named chr [1:3] "2" "1" "0"
## ..- attr(*, "names")= chr [1:3] "collective" "individual" "no protest"
```

```
## $ sexism : num 4.87 4.25 5 5.5 5.62 5.75 5.12 6.62 5.75 4.62 ...
## $ anger : num 2 1 3 1 1 1 2 1 6 1 ...
## $ liking : num 4.83 4.5 5.5 5.66 6.16 6 4.66 6.5 1 6.83 ...
## $ respappr: num 4.25 5.75 4.75 7 6.75 5.5 5 6.25 3 5.75 ...
## $ prot2 : num 1 0 1 1 1 1 1 0 0 0 ...
## ..- attr(*, "value.labels")= Named chr [1:3] "2" "1" "0"
## ..- attr(*, "names")= chr [1:3] "collective" "individual" "no protest"
```

## Nombre de las variables

```
colnames(x)
```

```
## [1] "protest" "sexism" "anger" "liking" "respappr" "prot2"
```

## Matriz de correlaciones

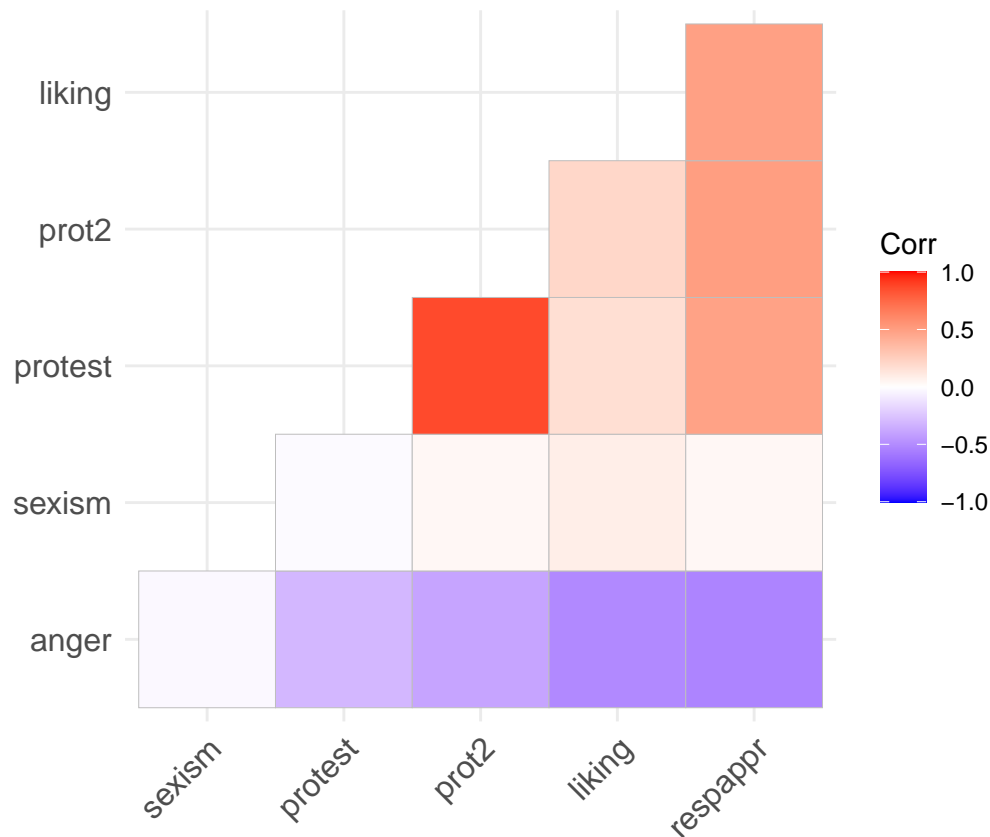
```
R <- hetcor(x)$correlations
```

```
R
```

```
##           protest      sexism      anger      liking      respappr
## protest  1.00000000 -0.02090619 -0.30768448  0.16930838  0.48488889
## sexism   -0.02090619  1.00000000 -0.03180676  0.09139386  0.04218515
## anger    -0.30768448 -0.03180676  1.00000000 -0.50538104 -0.53329781
## liking    0.16930838  0.09139386 -0.50538104  1.00000000  0.49440350
## respappr  0.48488889  0.04218515 -0.53329781  0.49440350  1.00000000
## prot2     0.86252472  0.04021842 -0.39218836  0.21313755  0.49920855
##
##           prot2
## protest  0.86252472
## sexism   0.04021842
## anger    -0.39218836
## liking    0.21313755
## respappr  0.49920855
## prot2     1.00000000
```

## Gráfico de correlaciones

```
ggcorrplot(R, type = "lower", hc.order = TRUE)
```



En este gráfico es posible apreciar las correlaciones entre nuestras variables

## Factorización de la matriz de correlaciones

```
p_Barlett <- cortest.bartlett(R)
```

```
## Warning in cortest.bartlett(R): n not specified, 100 used
```

## Visualización de p-valor

```
p_Barlett$p.value
```

```
## [1] 3.166949e-42
```

- Ho: Las variables están correlacionadas
- Ha: Las variables no están correlacionadas.

No se rechaza la hipótesis nula ya que nuestras variables están correlacionadas.

## Criterio Kaiser-Mayer

- 0,00 a 0.49 No adecuados
- 0.50 A 0.59 Poco adecuados
- 0.60 a 0.69 Aceptables
- 0.70 a 0.89 Buenos
- 0.80 a 1.00 Excelentes

KMO(R)

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = R)
## Overall MSA = 0.68
## MSA for each item =
##   protest   sexism   anger   liking respappr   prot2
##    0.60     0.30     0.76    0.72     0.82     0.62
```

Una vez obtuvimos el valor, podemos concluir que nuestros datos son “aceptables” para nuestro análisis factorial.

## Extracción de factores

- minres: mínim residuo
- mle: max Verosimilitud
- paf: ejes principales
- alpha: alfa
- minchi: minimos cuadrados
- minrak: minimo rango

Es importante recordar que los numeros cercanos a 1, significarán que el factor si explica a nuestras variables

```
modelo1 <- fa(R, nfactor = 3, rotate = "none", fm = "mle")
```

```
modelo2 <- fa(R, nfactor = 3, rotate = "none", fm = "minres")
```

## Extracción del resultado de las comunididades

Encontraremos la proporción de la varianza explicada. Se interpreta de tal forma que números cercanos a 1, el factor explicará mejor la variable.

```
C1 <- sort(modelo1$communality, decreasing = TRUE)
```

```
C2 <- sort(modelo2$communality, decreasing = TRUE)
```

```
head(cbind(C1, C2))
```

```
##           C1      C2
## prot2    0.99500020 0.9950667
## protest  0.99500003 0.9001687
## respappr 0.58057260 0.8383268
## liking   0.52857226 0.5793778
## anger    0.52404490 0.5324390
## sexism   0.01877234 0.5191249
```

Recordemos que los números cercanos a 1, significarán que el factor explica a la variable, por lo que, en este ejemplo nuestra variable 1 (prot2), al igual que nuestra variable 2 (protest) están muy bien explicadas.

## Extracción de unicidades

La unicidad es el cuadrado del coeficiente del factor único, y se expresa como la porción de la varianza explicada por el factor único. Es decir, no puede ser explicada por otros factores. 4

```
u1 <- sort(modelo1$uniquenesses, decreasing = TRUE)
```

```
u2 <- sort(modelo2$uniquenesses, decreasing = TRUE)
```

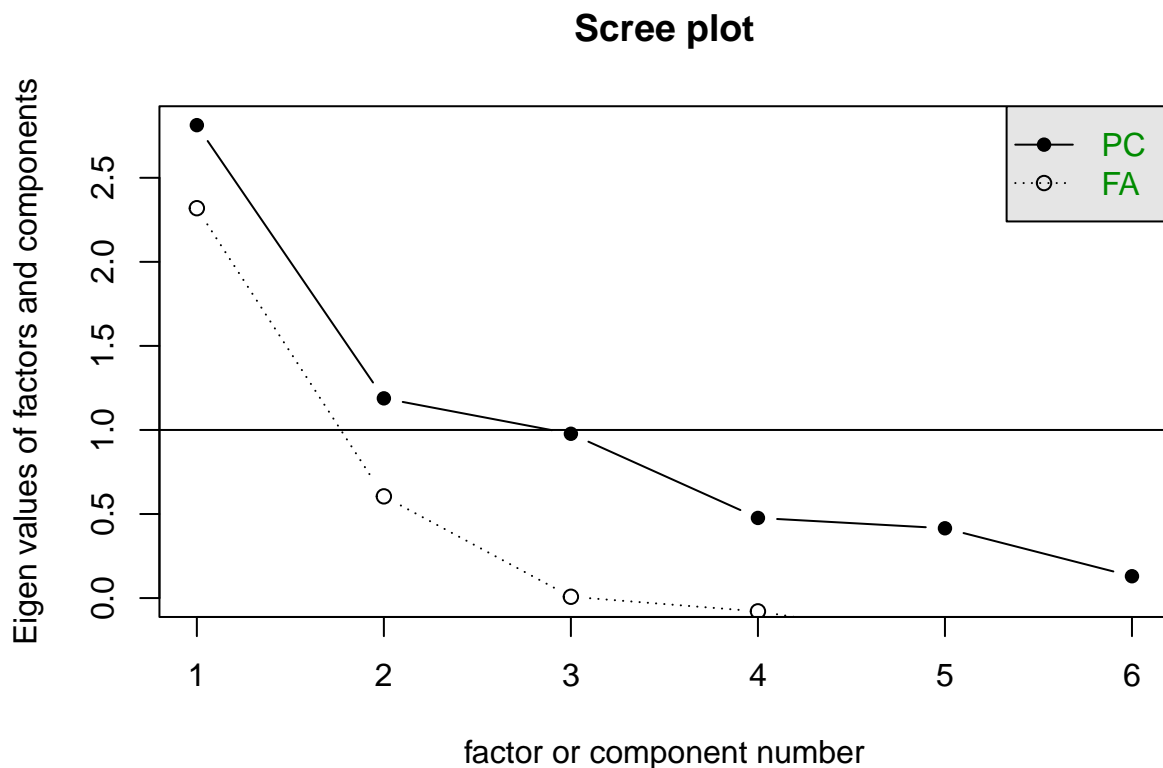
```
head(cbind(u1,u2))
```

```
##           u1           u2
## sexism  0.981227657 0.480875149
## anger   0.475955099 0.467561038
## liking  0.471427739 0.420622196
## respappr 0.419427396 0.161673215
## protest 0.004999965 0.099831348
## prot2   0.004999804 0.004933328
```

Una vez observando la tabla podemos concluir que las variables las cuales cuentan con valores cercanos a 1 son las cuales estan mejor explicadas, en este caso, sería nuestra variable “02”.

## Generación de screeplot

```
scree(R)
```



En este gráfico es posible observar la representación de los componentes principales y los factores principales, por lo que podemos decir que para este ejemplo, podríamos tomar 2 componentes principales y solo 1 factor principal.

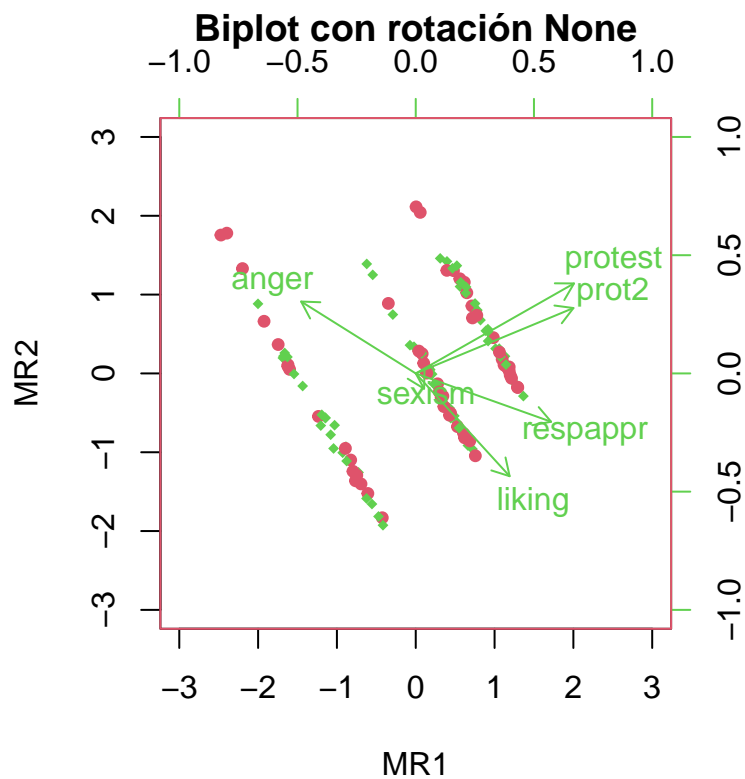
## Rotación de la matriz

```
install.packages("GPArotation")
library(GPArotation)
```

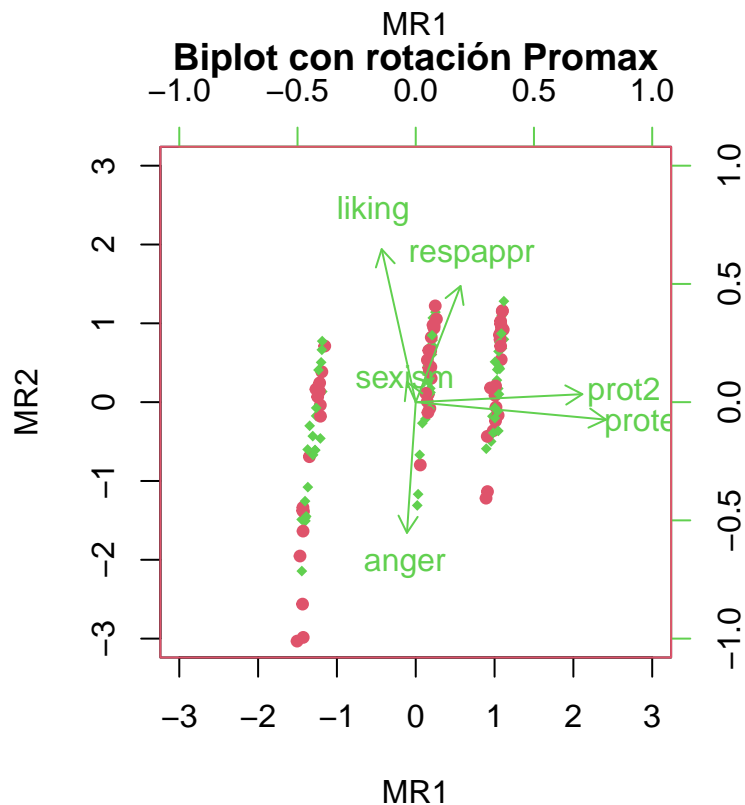
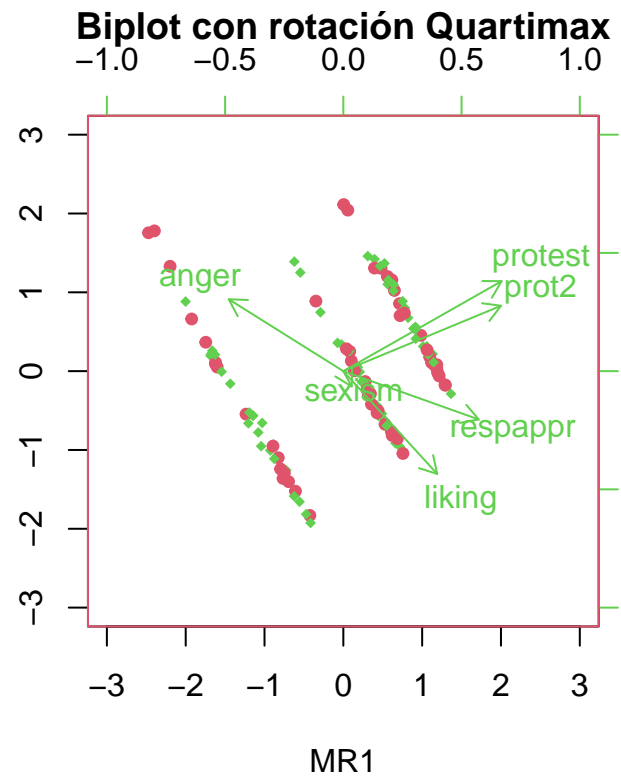
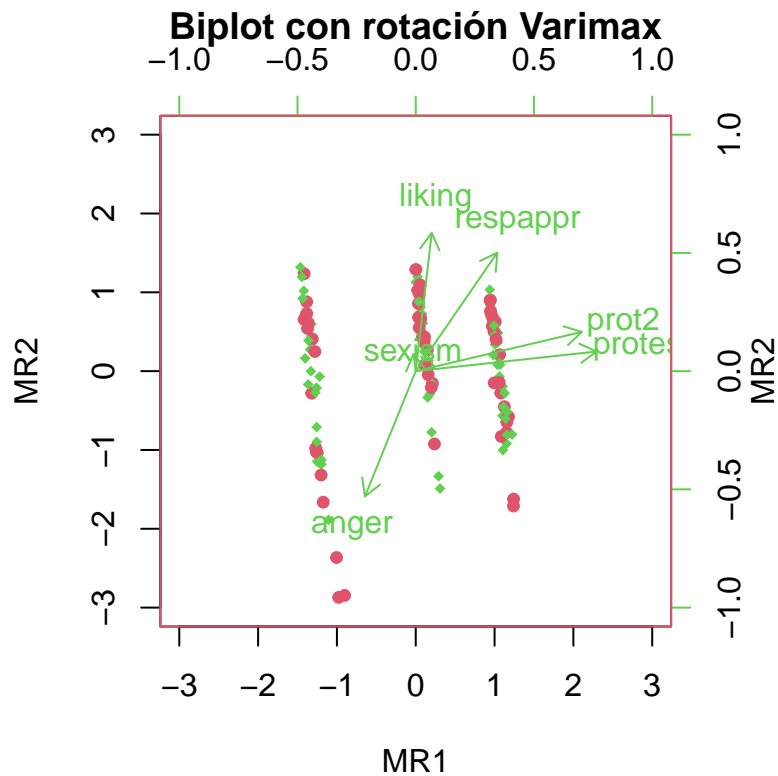
```
rot <- c("None", "Varimax", "Quartimax", "Promax")
bi_mod <- function(tipo){
  biplot.psych(fa(x, nfactors = 2,
    fm= "minres", rotate = tipo),
    main = paste("Biplot con rotación", tipo),
    col = c(2,3,4), pch = c(21,18), group = bfi[, "gender"])
}
```

```
sapply(rot, bi_mod)
```

```
## Specified rotation not found, rotate='none' used
```



```
## Specified rotation not found, rotate='none' used
```



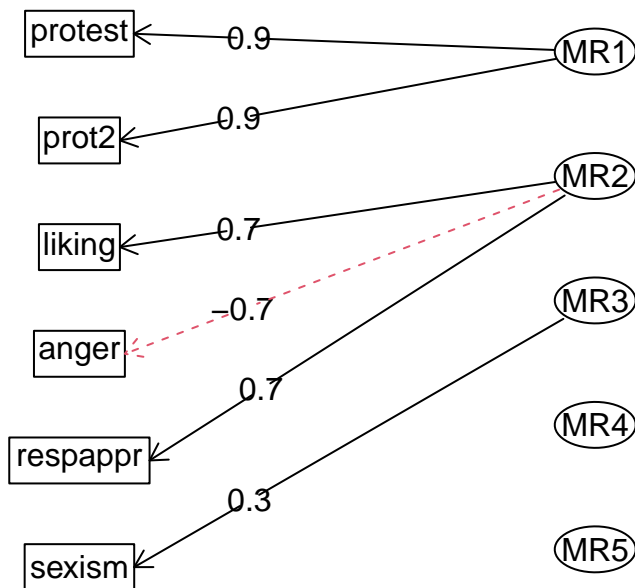
```
## $None
## NULL
##
## $Varimax
## NULL
```

```
##
## $Quartimax
## NULL
##
## $Promax
## NULL

#Grafico de arbol
modelo_varimax <- fa(R, nfactors = 5,
                    rotate = "varimax",
                    fm = "minres")

fa.diagram(modelo_varimax)
```

## Factor Analysis



## Vizualización de la matriz de carga rotada

```
print(modelo_varimax$loadings, cut= 0)
```

```
##
## Loadings:
##      MR1      MR2      MR3      MR4      MR5
## protest  0.925  0.186 -0.083  0.121  0.030
## sexism   -0.001  0.045  0.344  0.000  0.000
## anger    -0.212 -0.712 -0.001  0.167  0.033
## liking    0.044  0.716  0.173  0.090 -0.025
## respappr  0.384  0.654  0.039  0.061  0.115
## prot2     0.913  0.240  0.088 -0.159 -0.022
##
##      MR1      MR2      MR3      MR4      MR5
```



```
## SS loadings      1.885 1.542 0.164 0.080 0.016
## Proportion Var  0.314 0.257 0.027 0.013 0.003
## Cumulative Var  0.314 0.571 0.599 0.612 0.615
```

Por último, podemos comentar que las líneas rojas en el diagrama de árbol son las cargas negativas y las líneas negras son cargas positivas, de igual manera, viendo los resultados en la función “loadings” podemos ver como es que numericamente están agrupados nuestros factores.

En este caso, para la variable “protest” tiene mayor carga en el factor 1 en la segunda variable en el factor 2, la tercera en el 4, etc.