

Examen

Salazar Vega Rodrigo

14/12/2021

```
#knitr::opts_chunk$set(echo = TRUE)
library(nortest)
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.1.2

## Loading required package: ggplot2

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

library(ggplot2)
library(polycor)

## Warning: package 'polycor' was built under R version 4.1.2

library(ggcorrplot)
library(psych)

## Warning: package 'psych' was built under R version 4.1.2

##
## Attaching package: 'psych'

## The following object is masked from 'package:polycor':
##
##      polyserial

## The following objects are masked from 'package:ggplot2':
##
##      %+%, alpha

datos <- read.table(file.choose(), header=TRUE)
datos <- data.frame(datos)
datos_esc <- scale(datos)
```

Resolucion de examen.

Primero cargamos los datos al entorno de trabajo y lo transformamos a un data frame para poder trabajar más comodo don dichos datos, empezaremos con la prueba de hipotesis, ya que en esta podemos manipular los datos tal cual esta, en las demás pruebas es necesario escalar los datos, por eso empezaremos con esta, aplicaremos una prueba de normalidad.

```

cat("Exploracion de datos para ver si tiene una distribucion normal\n")
## Exploracion de datos para ver si tiene una distribucion normal
print("Anderson-Darling normality test\n")
## [1] "Anderson-Darling normality test\n"
print(ad.test(datos$laufkont))

##
## Anderson-Darling normality test
##
## data:  datos$laufkont
## A = 86.287, p-value < 2.2e-16

cat("\nComo el p-value es menor a 0.5 podemos descartar que tenga una
distribucion normal")

##
## Como el p-value es menor a 0.5 podemos descartar que tenga una
distribucion normal

print(t.test(datos$laufkont, alternative = "less", conf.level = 0.95))

##
## One Sample t-test
##
## data:  datos$laufkont
## t = 64.798, df = 999, p-value = 1
## alternative hypothesis: true mean is less than 0
## 95 percent confidence interval:
##      -Inf 2.642477
## sample estimates:
## mean of x
##      2.577

```

Analisis factorial.

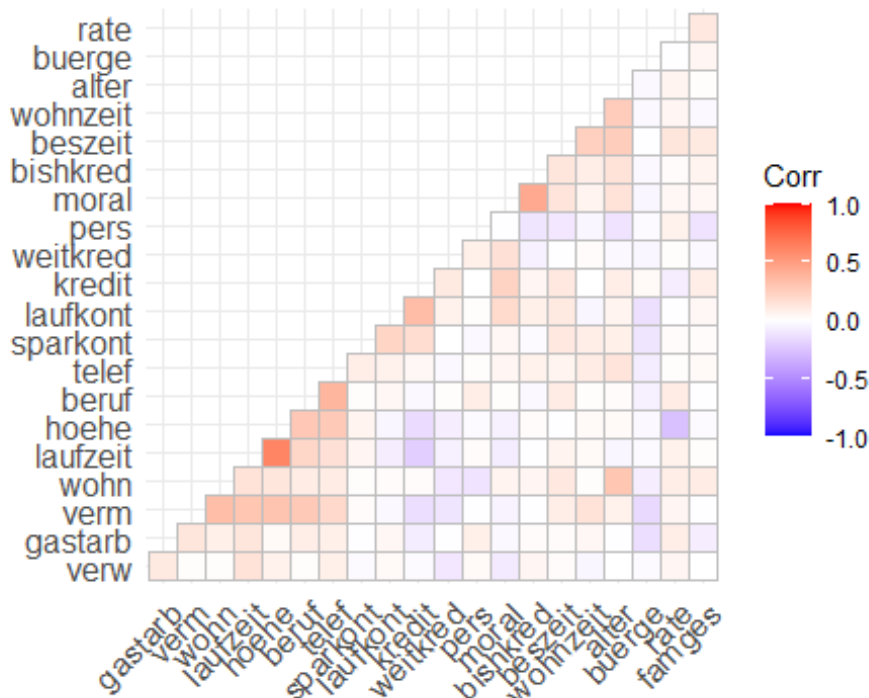
Como se habia mencionado anteriormente para trabajar esta técnica tenemos que tener los datos escalados, por lo se realiza eso y procedemos a realizar el análisis, utilizamos una funcion la que nos muestra cual es el número de factores optimos para trabajar con nuestro data set, el resultado son los siguientes:

```

# Obtenemos La matriz de correlacion policorica
mat_cor <- hetcor(datos_esc)$correlations #matriz de correlacion
policorica
plot(ggcorrplot(mat_cor, type="lower", hc.order = T, title = "Grafica
de correlaciones"))

```

Grafica de correlaciones



```
# Verificamos que la matriz sea factoriazble
cortest.bartlett(mat_cor, n = 100)->p_esf
cat("\nBartlett Test\n")

##
## Bartlett Test

print(p_esf$p)

## [1] 0.02134718

cat("\nKMO\n")

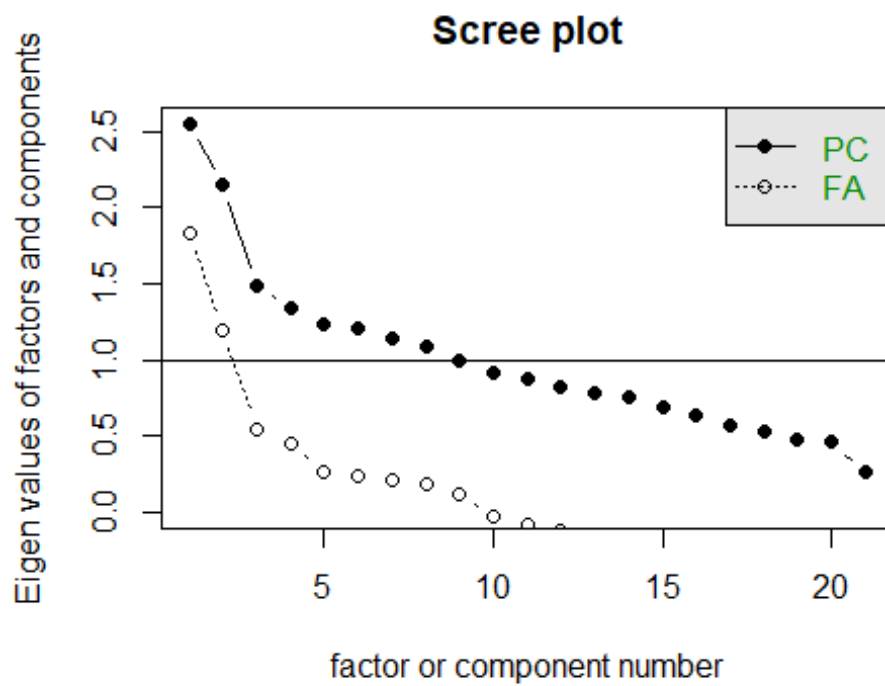
##
## KMO

print(KMO(mat_cor))

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = mat_cor)
## Overall MSA = 0.6
## MSA for each item =
## laufkont laufzeit moral verw hoehe sparkont beszeit
rate
## 0.65 0.57 0.58 0.53 0.54 0.63 0.70
0.32
## famges buerge wohnzeit verm alter weitkred wohn
bishkred
```

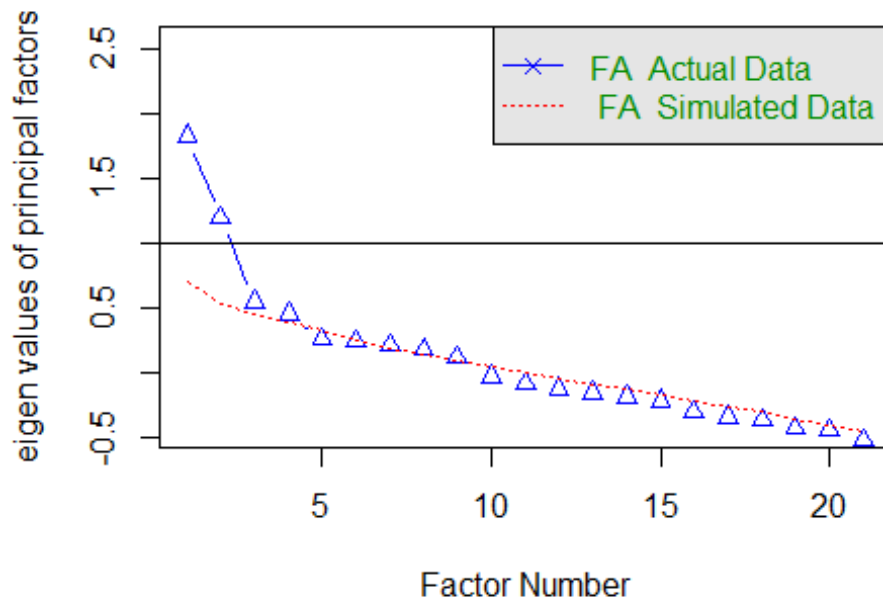
```
##      0.54      0.60      0.57      0.73      0.61      0.57      0.60
0.53
##      beruf      pers      telef      gastarb      kredit
##      0.66      0.61      0.70      0.66      0.66
```

```
# Determinar el numero de factores
plot(scee(mat_cor))
```



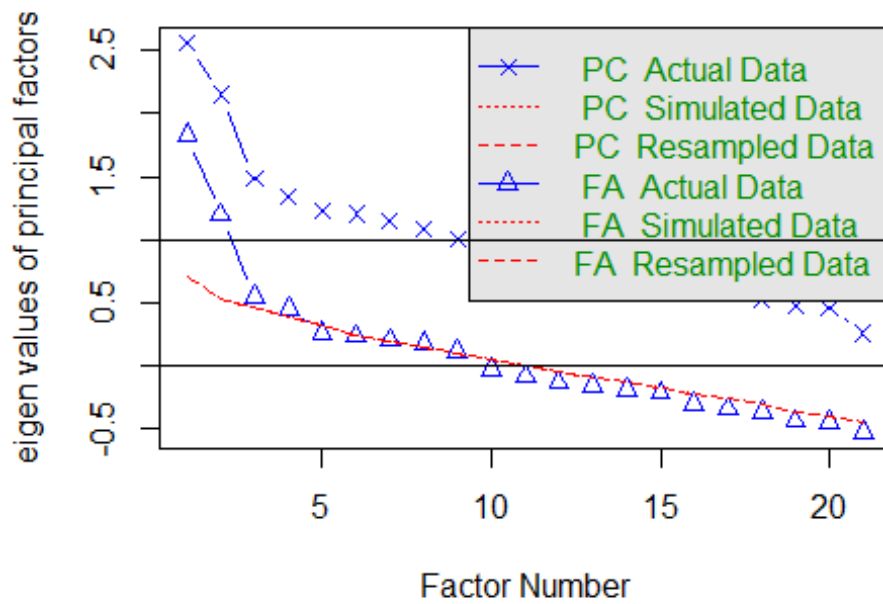
```
plot(fa.parallel(mat_cor,n.obs=200,fa="fa",fm="minres"))
```

Parallel Analysis Scree Plots



Parallel analysis suggests that the number of factors = 4 and the number of components = NA

Parallel Analysis Scree Plots



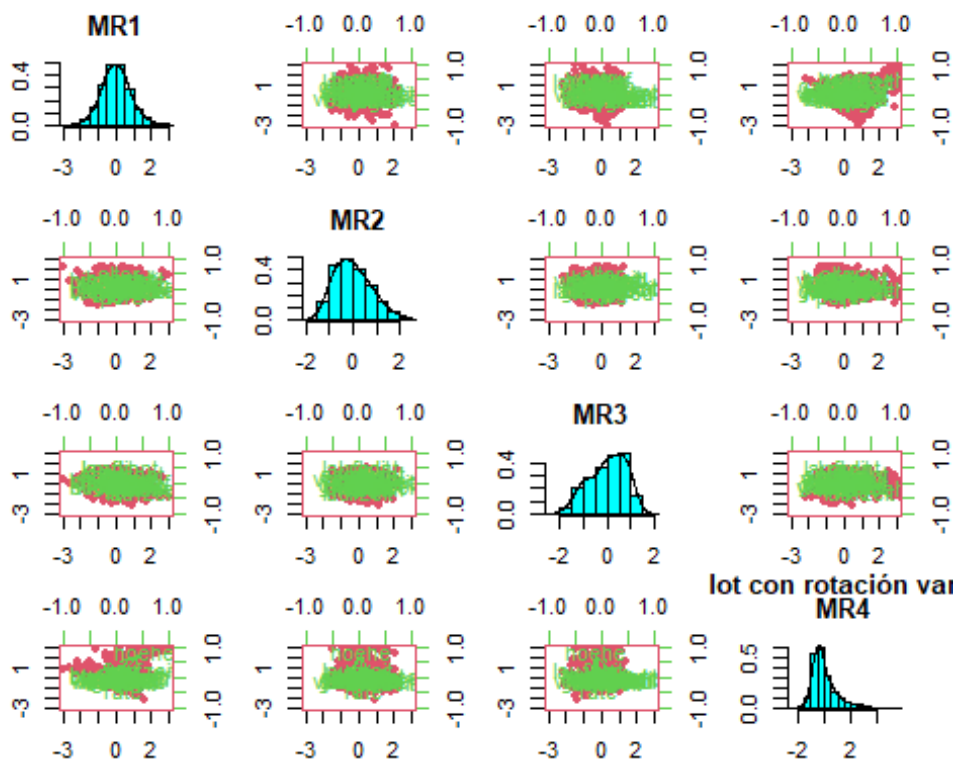
```

#Rotacion
rot<-c("varimax")

bi_mod<-function(tipo){
  biplot.psych(fa(datos_esc, nfactors = 4, fm = "minres", rotate =
tipo),main = paste("Biplot con rotación",tipo),col=c(2,3,4),pch =
c(21,18))
}
sapply(rot,bi_mod)

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =
rotate, : An
## ultra-Heywood case was detected. Examine the results carefully

```



```

## $varimax
## NULL

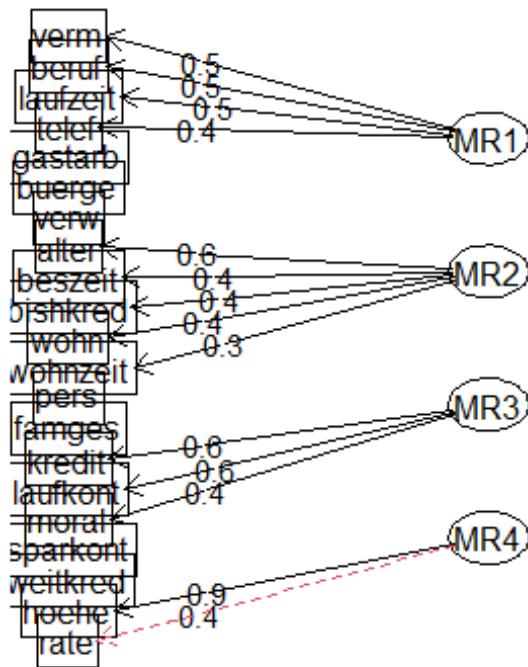
# Interpretacion
modelo_varimax<-fa(mat_cor,nfactors = 4,rotate = "varimax",
fa="minres")

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =
rotate, : An
## ultra-Heywood case was detected. Examine the results carefully

fa.diagram(modelo_varimax, main = "Gráfico AF")

```

Gráfico AF



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

```
##
## Loadings:
##      MR1      MR2      MR3      MR4
## laufkont  0.080  0.057  0.567 -0.019
## laufzeit  0.518  0.016 -0.211  0.316
## moral     -0.082  0.345  0.363  0.022
## verw      0.136 -0.026 -0.034  0.002
## hoehe     0.528  0.033 -0.126  0.855
## sparkont  0.108  0.089  0.265  0.032
## beszeit   0.120  0.419  0.112 -0.089
## rate      0.166  0.104 -0.044 -0.390
## famges    -0.019  0.154  0.036 -0.017
## buerge    -0.205 -0.012 -0.111  0.073
## wohnzeit   0.090  0.327 -0.028 -0.046
## verm      0.533  0.180 -0.179  0.016
## alter      0.064  0.556  0.039 -0.040
## weitkred  -0.059 -0.073  0.207 -0.031
## wohn       0.275  0.360 -0.088 -0.034
## bishkred  -0.059  0.375  0.134  0.069
## beruf      0.521 -0.006  0.078  0.019
## pers       0.129 -0.276  0.083 -0.097
## telef      0.423  0.119  0.133  0.087
## gastarb    0.279 -0.033 -0.024 -0.134
## kredit    -0.139  0.120  0.571 -0.007
##
```

```
##          MR1    MR2    MR3    MR4
## SS loadings  1.619 1.173 1.073 1.045
## Proportion Var 0.077 0.056 0.051 0.050
## Cumulative Var 0.077 0.133 0.184 0.234
```

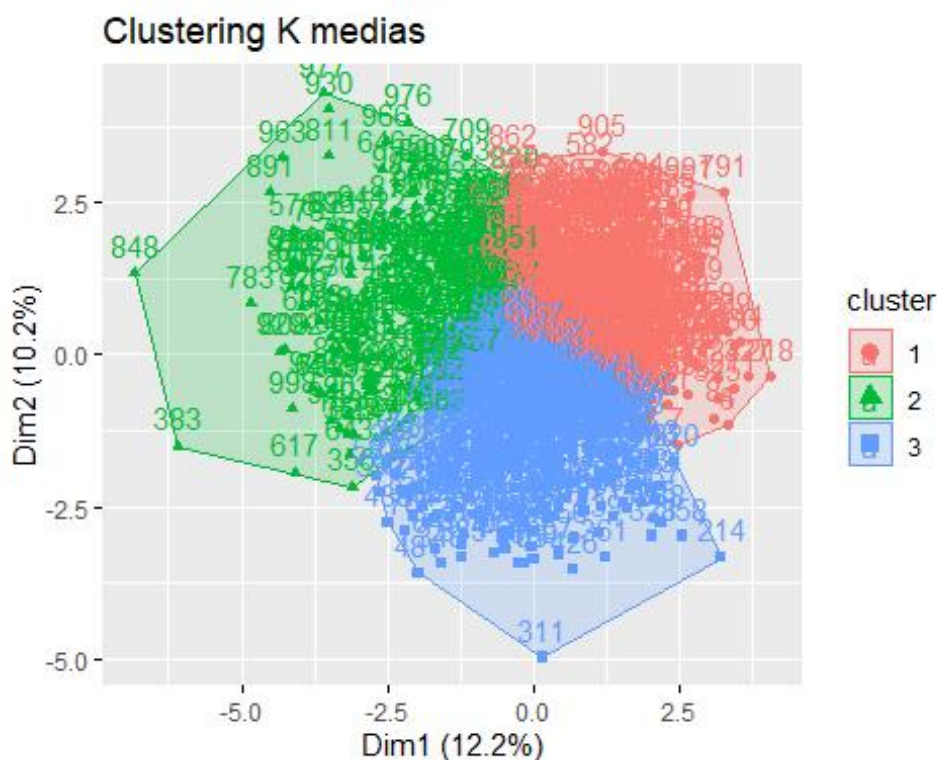
Obtenemos la matriz de correlaciones, así como el gráfico donde muestra el número de factores requeridos y la relación que tiene las variables con cada factor.

k-means

Para trabajar con el k-means nos guiamos un poco con el análisis anterior, definimos 3 grupos para trabajar con este análisis y obtenemos el siguiente resultado.

```
#se aplica el algoritmo k-means
grupos <- kmeans(datos_esc, centers = 3, nstart = 25)

#Graficar los grupos
plot(fviz_cluster(grupos, data = datos_esc, main = "Clustering K
medias"))
```



```
relaciones <- data.frame(grupos_km = grupos$cluster)
```