

Escalamiento no métrico

2024-04-24

Librerias necesarias

```
library(car)

## Loading required package: carData
library(smacof)

## Loading required package: plotrix
## Loading required package: colorspace
## Loading required package: e1071
##
## Attaching package: 'smacof'
## The following object is masked from 'package:base':
##       transform
library(cluster)
library(lubridate)

##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
##       date, intersect, setdiff, union
library(andrews)

## See the package vignette with `vignette("andrews")` 
library(dplyr)

##
## Attaching package: 'dplyr'
## The following object is masked from 'package:car':
##       recode
## The following objects are masked from 'package:stats':
##       filter, lag
## The following objects are masked from 'package:base':
##       intersect, setdiff, setequal, union
```

```

library(corrplot)

## corrplot 0.92 loaded
source("utilerias/funciones.R")

```

Introducción:

Se hace un análisis de escalamiento multidimensional, seleccionaremos la mejor dimensión con los ratings(columnas 5:18) de la base de datos RockHard del paquete **smacof**. Los datos son de la revista RockHard, una revista alemana de heavy metal, los redactores valoran cada mes alrededor de 50 discos en una escala de (0... peor a 10... mejor), el conjunto de datos contiene todas las calificaciones de 2013. Los evaluadores en las columnas, y las bandas/álbumes en las filas.

Sinopsis

El escalamiento multidimensional no métrico tiene por objetivo preservar las disimilaridades mientras se posicionan los objetos en una menor dimension. Se aplica principalmente sobre datos ordinales o ratings.

Pasos esenciales:

- 0.- Preprocesamiento(Datos atípicos, escalamiento, transformaciones)
- 1.- Matriz de datos con escala ordinales (Definir rangos)
- 2.- Cálculo de disimilaridades
- 4.- Escalamiento no métrico (Regresión monótona)
- 5.- Mejoras

Carga de la informacion

Práctica sobre el los ratings de bandas y álbumes de música de la revista RockHard para los 12 meses del año 2013, un album por cada banda.

Variables: Indices o ratings dados por 14 personas: Gotz, Thomas, Frank, Bjorn, Jan, Boris, Himmelstein, Michael, Jens, Ronny, Felix, Jakob, Marcus y Jenny.

Mostramos los primeros 10 banda y álbum.

```

data1 <- smacof:::RockHard
head(data1, 10)

```

	##	Year	Month	Band	Album	Götz	Thomas						
## 1	2013	1		Attic	The Invocation	8.5	8.0						
## 2	2013	1		Paradox	Tales Of The Weird	7.5	7.0						
## 3	2013	1		Züül	To The Frontlines	8.0	7.0						
## 4	2013	1	Chapel Of Disease	Summoning	Black Gods	8.0	7.5						
## 5	2013	1	Dropkick Murphys	Signed And Sealed In Blood		7.0	7.5						
## 6	2013	1		Saturnus	Saturn In Ascension	6.0	7.0						
## 7	2013	1	Golden Void		Thrill 318	7.5	7.5						
## 8	2013	1	Fragments Of Unbecoming	The Art Of Coming Apart		8.0	7.0						
## 9	2013	1			Stamina	7.5	6.5						
## 10	2013	1	After Oblivion	Arkham Witch	Legions Of The Deep	8.0	7.0						
	##	Frank	Björn	Jan	Boris	Himmelstein	Michael	Jens	Ronny	Felix	Jakob	Marcus	
## 1		8.0	9.0	7.0	8.5		8.5	7.0	NA	NA	NA	8.5	7.5
## 2		8.0	9.0	7.0	7.5		7.0	6.5	NA	NA	NA	7.5	8.0

```

## 3 7.5 8.5 7.0 8.5      8.0 6.5 NA NA NA 8.0 7.0
## 4 8.0 8.0 7.5 8.0      8.0 6.5 NA NA NA 8.0 6.5
## 5 7.5 6.0 6.5 7.0      8.5 8.5 NA NA NA 7.5 7.0
## 6 6.5 8.0 7.0 7.5      7.0 6.0 NA NA NA 8.5 7.5
## 7 6.0 7.5 6.5 7.5      8.0 7.5 NA NA NA 7.0 8.0
## 8 8.0 8.5 6.0 7.0      7.0 6.5 NA NA NA 7.0 6.5
## 9 8.5 7.0 7.5 7.0      7.0 6.0 NA NA NA 7.0 6.5
## 10 8.0 6.5 6.5 7.5     7.5 6.0 NA NA NA 7.0 7.0

## Jenny
## 1 7.0
## 2 7.5
## 3 6.5
## 4 6.0
## 5 6.5
## 6 7.0
## 7 5.0
## 8 6.5
## 9 7.0
## 10 6.0

```

Formato Correcto

Primero creamos la variable Band_Album y luego la establecemos como índice de la base de datos. Quitamos variables que no usaremos en el análisis. Se muestran las primeras 10 observaciones.

```

data1$Band_Album<-paste(data1$Band, "_", data1$Album)

rownames(data1) <- data1$Band_Album # Estableciendo como indice las bandas
data1$Band_Album <- NULL # Estableciendo como indice las bandas
data1 <- subset(data1, select = -c(Year, Month, Band, Album))
head(data1)

##                                     Götz Thomas Frank Björn Jan Boris
## Attic _ The Invocation           8.5    8.0    8.0   9.0 7.0 8.5
## Paradox _ Tales Of The Weird    7.5    7.0    8.0   9.0 7.0 7.5
## Züül _ To The Frontlines        8.0    7.0    7.5   8.5 7.0 8.5
## Chapel Of Disease _ Summoning Black Gods 8.0    7.5    8.0   8.0 7.5 8.0
## Dropkick Murphys _ Signed And Sealed In Blood 7.0    7.5    7.5   6.0 6.5 7.0
## Saturnus _ Saturn In Ascension 6.0    7.0    6.5   8.0 7.0 7.5
##                                     Himmelstein Michael Jens Ronny
## Attic _ The Invocation           8.5    7.0    NA    NA   NA
## Paradox _ Tales Of The Weird    7.0    6.5    NA    NA   NA
## Züül _ To The Frontlines        8.0    6.5    NA    NA   NA
## Chapel Of Disease _ Summoning Black Gods 8.0    6.5    NA    NA   NA
## Dropkick Murphys _ Signed And Sealed In Blood 8.5    8.5    NA    NA   NA
## Saturnus _ Saturn In Ascension 7.0    6.0    NA    NA   NA
##                                     Felix Jakob Marcus Jenny
## Attic _ The Invocation           NA    8.5    7.5    7.0
## Paradox _ Tales Of The Weird    NA    7.5    8.0    7.5
## Züül _ To The Frontlines        NA    8.0    7.0    6.5
## Chapel Of Disease _ Summoning Black Gods  NA   8.0    6.5    6.0
## Dropkick Murphys _ Signed And Sealed In Blood  NA   7.5    7.0    6.5
## Saturnus _ Saturn In Ascension  NA   8.5    7.5    7.0

```

Selección de las columnas auxiliares y de análisis.

En esta sección analizamos algunos datos y generaremos otros. Por ejemplo definimos la variable `avrating` del promedio de los ratings de todos los participantes. Como hay muchos valores faltantes (NA's), primero hacemos una revisión general para cada participante.

Podemos observar en la siguiente salida que los participantes que tienen más del 50% de NA's, son Felix y Jenny que tienen 478 y 383 NA'S de las 576 observaciones de la base de datos, respectivamente.

```
sprintf("Gotz: %d de %d", sum(is.na(data1$Götz)), 576)
```

```
## [1] "Gotz: 0 de 576"
```

```
sprintf("Thomas: %d de %d", sum(is.na(data1$Thomas)), 576)
```

```
## [1] "Thomas: 0 de 576"
```

```
sprintf("Frank: %d de %d", sum(is.na(data1$Frank)), 576)
```

```
## [1] "Frank: 0 de 576"
```

```
sprintf("Björn: %d de %d", sum(is.na(data1$Björn)), 576)
```

```
## [1] "Björn: 0 de 576"
```

```
sprintf("Jan: %d de %d", sum(is.na(data1$Jan)), 576)
```

```
## [1] "Jan: 0 de 576"
```

```
sprintf("Boris: %d de %d", sum(is.na(data1$Boris)), 576)
```


[1] "Boris: 0 de 576"

```
sprintf("Himmelstein: %d de %d", sum(is.na(data1$Himmelstein)), 576)
```

```
## [1] "Himmelstein: 0 de 576"
```

```
sprintf("Michael: %d de %d", sum(is.na(data1$Michael)), 576)
```

```
## [1] "Michael: 52 de 576"
```

```
sprintf("Jens: %d de %d", sum(is.na(data1$Jens)), 576)
```

```
## [1] "Jens: 193 de 576"
```

```
sprintf("Ronny: %d de %d", sum(is.na(data1$Ronny)), 576)
```

```
## [1] "Ronny: 282 de 576"
```

```
sprintf("Felix: %d de %d", sum(is.na(data1$Felix)), 576)
```

```
## [1] "Felix: 478 de 576"
```

```
sprintf("Jakob: %d de %d", sum(is.na(data1$Jakob)), 576)
```

```
## [1] "Jakob: 98 de 576"
```

```
sprintf("Marcus: %d de %d", sum(is.na(data1$Marcus)), 576)
```

```
## [1] "Marcus: 242 de 576"
```

```
sprintf("Jenny: %d de %d", sum(is.na(data1$Jenny)), 576)
```

```
## [1] "Jenny: 383 de 576"
```

Observemos en la siguiente tabla de correlaciones, que todas están en los mismos órdenes, por lo que tomaremos las primeras 7 variables y omitiremos las últimas 7, pues generan muchos problemas al tener NA's y excluirlos no afecta realmente las relaciones.

```
#cor(na.omit(data1))
options(digits=2)
cor(data1, method = "pearson", use = "pairwise.complete.obs")
```

	Götz	Thomas	Frank	Björn	Jan	Boris	Himmelstein	Michael	Jens	Ronny	
## Götz	1.00	0.57	0.49	0.53	0.40	0.63		0.57	0.32	0.51	0.37
## Thomas	0.57	1.00	0.38	0.42	0.32	0.71		0.63	0.45	0.48	0.40
## Frank	0.49	0.38	1.00	0.50	0.45	0.35		0.32	0.33	0.54	0.46
## Björn	0.53	0.42	0.50	1.00	0.40	0.48		0.39	0.30	0.44	0.40
## Jan	0.40	0.32	0.45	0.40	1.00	0.33		0.30	0.28	0.36	0.32
## Boris	0.63	0.71	0.35	0.48	0.33	1.00		0.68	0.38	0.45	0.30
## Himmelstein	0.57	0.63	0.32	0.39	0.30	0.68		1.00	0.33	0.41	0.39
## Michael	0.32	0.45	0.33	0.30	0.28	0.38		0.33	1.00	0.43	0.42
## Jens	0.51	0.48	0.54	0.44	0.36	0.45		0.41	0.43	1.00	0.61
## Ronny	0.37	0.40	0.46	0.40	0.32	0.30		0.39	0.42	0.61	1.00
## Felix	0.39	0.43	0.27	0.43	0.26	0.25		0.52	0.20	0.26	0.24
## Jakob	0.51	0.57	0.41	0.47	0.42	0.59		0.49	0.46	0.55	0.48
## Marcus	0.38	0.51	0.28	0.22	0.24	0.42		0.38	0.49	0.48	0.34
## Jenny	0.49	0.44	0.40	0.37	0.32	0.46		0.34	0.39	NA	0.45
##	Felix	Jakob	Marcus	Jenny							
## Götz	0.39	0.51	0.38	0.49							
## Thomas	0.43	0.57	0.51	0.44							
## Frank	0.27	0.41	0.28	0.40							
## Björn	0.43	0.47	0.22	0.37							
## Jan	0.26	0.42	0.24	0.32							
## Boris	0.25	0.59	0.42	0.46							
## Himmelstein	0.52	0.49	0.38	0.34							
## Michael	0.20	0.46	0.49	0.39							
## Jens	0.26	0.55	0.48	NA							
## Ronny	0.24	0.48	0.34	0.45							
## Felix	1.00	NA	NA	NA							
## Jakob		NA	1.00	0.42	0.35						
## Marcus		NA	0.42	1.00	0.53						
## Jenny		NA	0.35	0.53	1.00						

En la siguiente tabla se muestran estas mismas correlaciones para las variables elegidas (participantes).

```
data1<-subset(data1, select = -c(Michael, Jens, Ronny, Felix, Jakob, Marcus, Jenny))
```

```
#cor(na.omit(data1))
options(digits=2)
cor(data1, method = "pearson", use = "pairwise.complete.obs")
```

	Götz	Thomas	Frank	Björn	Jan	Boris	Himmelstein
## Götz	1.00	0.57	0.49	0.53	0.40	0.63	0.57
## Thomas	0.57	1.00	0.38	0.42	0.32	0.71	0.63
## Frank	0.49	0.38	1.00	0.50	0.45	0.35	0.32
## Björn	0.53	0.42	0.50	1.00	0.40	0.48	0.39
## Jan	0.40	0.32	0.45	0.40	1.00	0.33	0.30
## Boris	0.63	0.71	0.35	0.48	0.33	1.00	0.68
## Himmelstein	0.57	0.63	0.32	0.39	0.30	0.68	1.00

Decidimos crear la variable `avrating`, que es el promedio de todas las calificaciones asignadas por todos los

7 participantes, considerando que hay NA's. Además, generamos la variable `grado`, que es el grado medido como bajo (0 a 4), medio (4 a 6), alto (6 a 8) y muy alto (8 a 10).

```
data1$avrating <- rowMeans(data1[, 1:7], na.rm=TRUE)

data1<-data1 %>% mutate(grado = case_when(avrating <= 4 ~ "Bajo",
                                             avrating <= 6 ~ "Medio",
                                             avrating <= 8 ~ "Alto",
                                             avrating <= 10 ~ "Muy alto"))

head(data1, 10)
```

	Götz	Thomas	Frank	Björn	Jan
## Attic _ The Invocation	8.5	8.0	8.0	9.0	7.0
## Paradox _ Tales Of The Weird	7.5	7.0	8.0	9.0	7.0
## Züül _ To The Frontlines	8.0	7.0	7.5	8.5	7.0
## Chapel Of Disease _ Summoning Black Gods	8.0	7.5	8.0	8.0	7.5
## Dropkick Murphys _ Signed And Sealed In Blood	7.0	7.5	7.5	6.0	6.5
## Saturnus _ Saturn In Ascension	6.0	7.0	6.5	8.0	7.0
## Golden Void _ Thrill 318	7.5	7.5	6.0	7.5	6.5
## Fragments Of Unbecoming _ The Art Of Coming Apart	8.0	7.0	8.0	8.5	6.0
## After Oblivion _ Stamina	7.5	6.5	8.5	7.0	7.5
## Arkham Witch _ Legions Of The Deep	8.0	7.0	8.0	6.5	6.5
##	Boris	Himmelstein	avrating		
## Attic _ The Invocation	8.5		8.5		8.2
## Paradox _ Tales Of The Weird	7.5		7.0		7.6
## Züül _ To The Frontlines	8.5		8.0		7.8
## Chapel Of Disease _ Summoning Black Gods	8.0		8.0		7.9
## Dropkick Murphys _ Signed And Sealed In Blood	7.0		8.5		7.1
## Saturnus _ Saturn In Ascension	7.5		7.0		7.0
## Golden Void _ Thrill 318	7.5		8.0		7.2
## Fragments Of Unbecoming _ The Art Of Coming Apart	7.0		7.0		7.4
## After Oblivion _ Stamina	7.0		7.0		7.3
## Arkham Witch _ Legions Of The Deep	7.5		7.5		7.3
##	grado				
## Attic _ The Invocation	Muy alto				
## Paradox _ Tales Of The Weird	Alto				
## Züül _ To The Frontlines	Alto				
## Chapel Of Disease _ Summoning Black Gods	Alto				
## Dropkick Murphys _ Signed And Sealed In Blood	Alto				
## Saturnus _ Saturn In Ascension	Alto				
## Golden Void _ Thrill 318	Alto				
## Fragments Of Unbecoming _ The Art Of Coming Apart	Alto				
## After Oblivion _ Stamina	Alto				
## Arkham Witch _ Legions Of The Deep	Alto				

Estableceremos la escala ordinal para la variable de grado.

```
# Establecimiendo de escalas ordinales
data1$grado <- factor(data1$grado, levels= c("Bajo", "Medio", "Alto", "Muy alto"), order=TRUE)
```

Luego indicamos las variables auxiliares, variables de análisis, y juntamos toda la información a utilizar en el análisis en una sola base de datos.

```
auxiliares <- colnames(data1[, c(8,9)])
analisis <- colnames(data1[,1:7]) # Seleccion de columnas
columnas <- c(auxiliares, analisis)
datos <- data1[, columnas] # Extraccion
```

Escalas Iniciales

Verificamos y asignamos las escalas iniciales, y el tipo de variables entre numéricas contínuas y enteras, o categóricas nominales y ordinales.

```
# Escalas
tipo <- sapply(datos, class)
continuas <- which(tipo == "numeric") # continuas
enteras <- which(tipo == "integer") # enteras
numericas <- names(c(continuas,enteras))

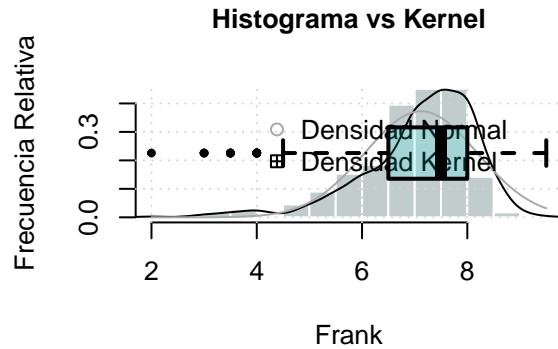
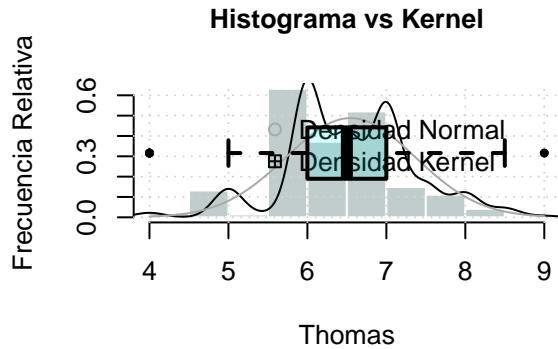
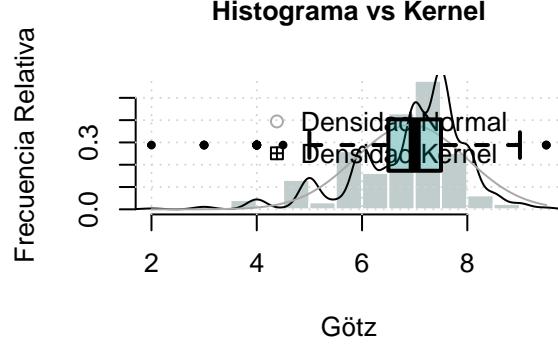
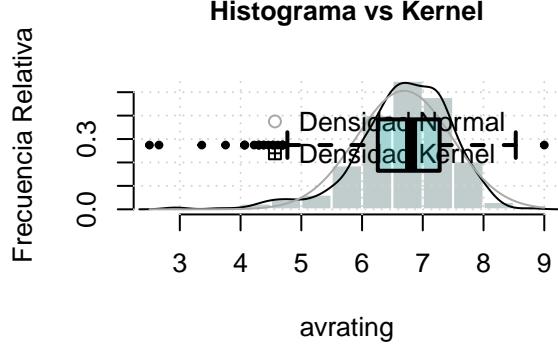
# Variables Categóricas
nominales <- which( tipo == "factor") # categóricas
ordinales <- which( sapply(datos, is.ordered) ) # ordinales
fecha <- which(tipo == "Date") # Fecha
categoricas <- names(c(nominales, ordinales, fecha))
```

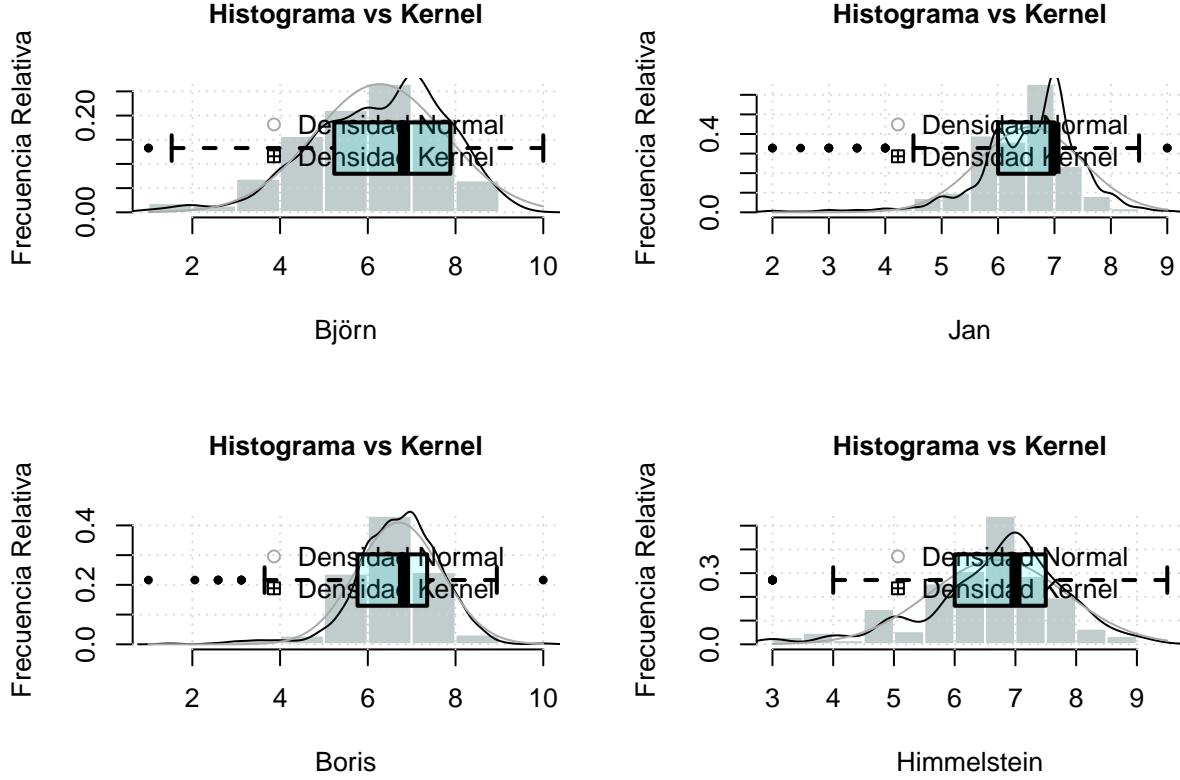
Descriptivos Multivariados

- Identificar Atípicos
- Problemas de escala
- Distribuciones

A continuación mostramos los histogramas, de los ratings de cada uno de los 14 participantes.

```
# Histogramas
par(mfrow= c(2,2) )
multi.hist(datos[, numericas])
```

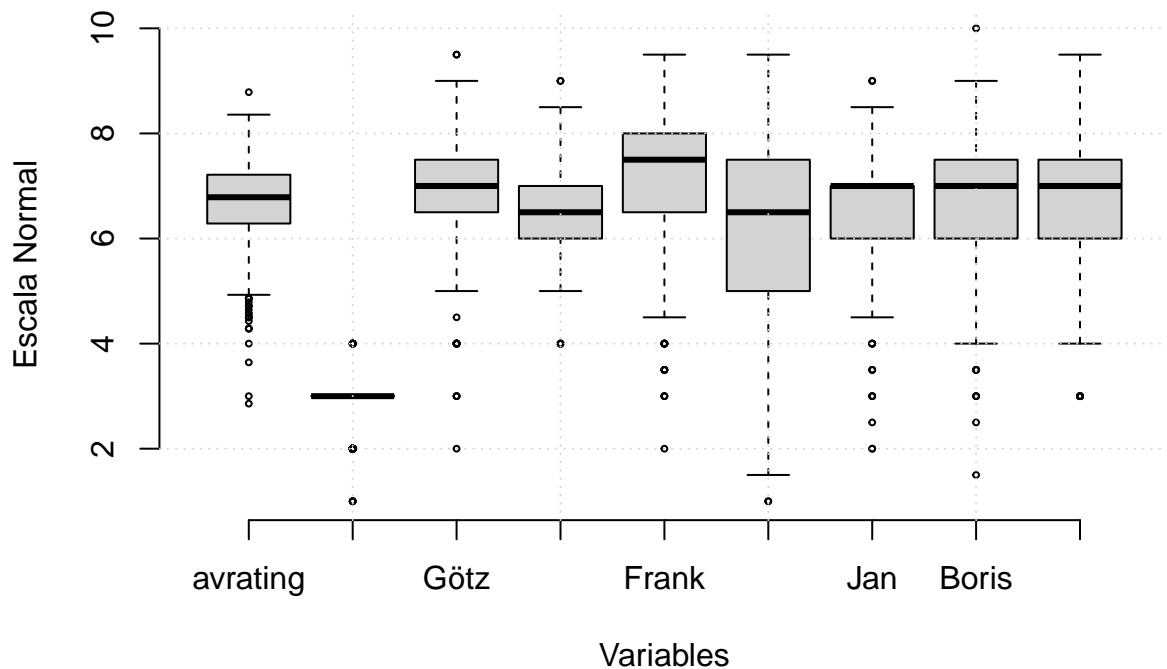




En el siguiente BoxPlot no parece haber problemas de escala, por lo que usaremos los datos sin estandarizar.

```
# Boxplot
boxplot(datos, main="Caja y Bigotes",
         frame = FALSE, xlab="Variables", ylab= "Escala Normal", cex=0.4);grid()
```

Caja y Bigotes



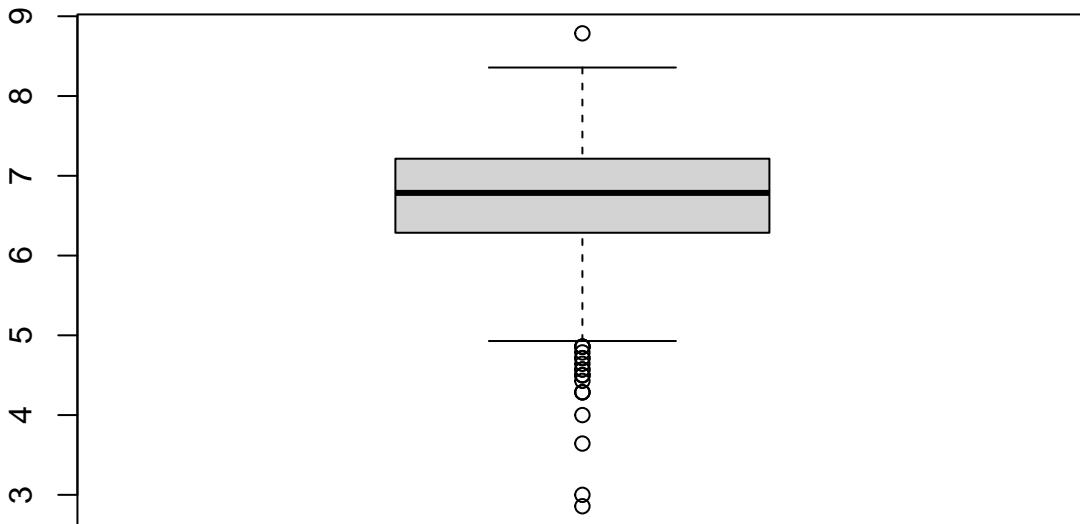
```
# Andrews ##CORREGIR!!!
#andrews(df = datos, type=2, bty = "n", ylab="f(t)", xlab="t", lwd=1, main="Grafico Andrews" ); grid()
```

Eliminacion de datos atípicos

- Importante ver que la variable auxiliar ayuda a identificar observaciones que afecten el análisis.

En el siguiente BoxPlot se observan algunos outliers por debajo del primer quartil, sin embargo no quitaremos esta información porque quita representatividad a la clasificación de “bajo”.

```
outliers <- boxplot(datos$avrating)$out
```



```
elementos <- which(datos$avrating %in% outliers)

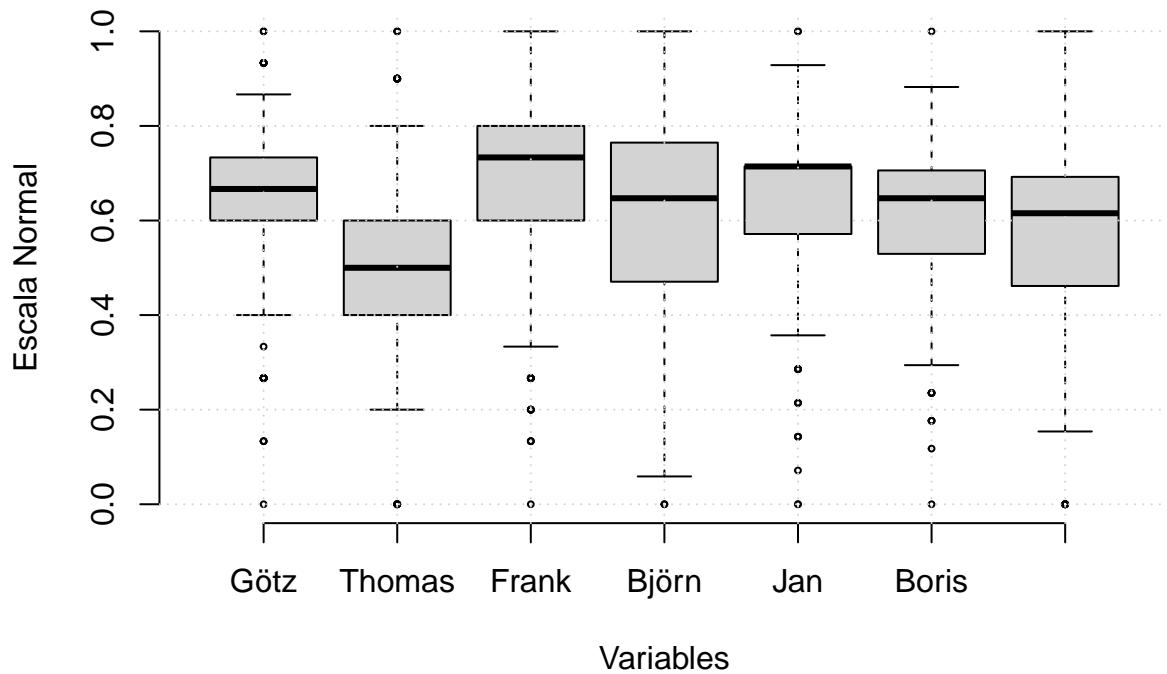
#datos <- datos[-union(elementos,elementos), ]
```

Escalamiento

Importante que los indices se recodifiquen a una escala ordinal, pero primero se normalizan ya que se trata de un indice.

```
# Normalizacion
datos[,analisis] <- sapply(datos[, analisis], function(data){
  (data - min(data, na.rm = TRUE)) / (max(data, na.rm = TRUE) - min(data, na.rm = TRUE)))}
# Boxplot
boxplot(datos[, analisis], main="Caja y Bigotes",
        frame = FALSE, xlab="Variables", ylab= "Escala Normal", cex=0.4);grid()
```

Caja y Bigotes



1 Matriz de datos con escala ordinales

Definicion de rangos para la escala de lickert

0-20 -> 1 21:40 -> 2 40:60 -> 3 61:80 -> 4 81:100 -> 5

```
# Transformacion a escala ordinal
datos[, analisis] <- datos[, analisis]*100
datos[, analisis] <- round(datos[, analisis])

for(indice in analisis){
  for(n in 1:nrow(datos)){
    datos[n,indice] = car::recode(datos[n,indice], "0:40=1; 41:60=2; 61:80=3; 81:100=4")
  }
}

# Formato Correcto
for(indice in analisis){
  datos[, indice] <- factor(datos[, indice], order = TRUE)
}

# Redefinicion de Escalas
tipo <- sapply(datos, class)
continuas <- which(tipo == "numeric") # continuas
enteras <- which(tipo == "integer") # enteras
numericas <- names(c(continuas,enteras))
```

```
# Variables Categoricas
nominales <- which( tipo == "factor") # categoricas
ordinales <- which( sapply(datos, is.ordered) ) # ordinales
fecha <- which(tipo == "Date") # Fecha
categoricas <- names(c(nominales, ordinales, fecha))
```

Calculo de la matriz de Disimilaridad

- Como las variables son en escala ordinal, ent se utiliza distancia gower(mixtas).

```
gower_dist <- daisy(datos[, analisis], metric = "gower")
```

Escalamiento no métrico

- Métricas de ajuste: En este caso, a prueba y error se encontro que 5 es la mejor dimensión.

Stress: con valor entre [0,1] y entre mas pequeño mejor.

RSS: entre mas pequeño mejor.

```
fit.datos <- smacofSym(gower_dist, type = "ordinal", ndim = 5)
Stress<-fit.datos$stress
sprintf("Stress: %f", Stress)
```

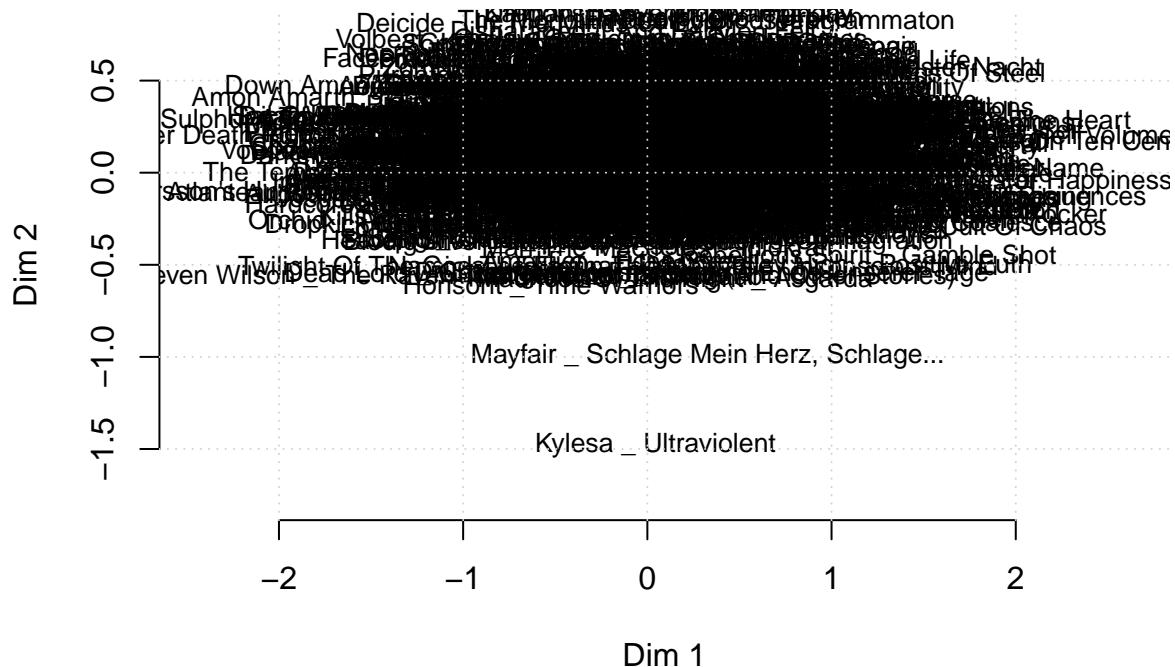
```
## [1] "Stress: 0.054728"
```

```
RSS<-fit.datos$rss
sprintf("RSS: %f", RSS)
```

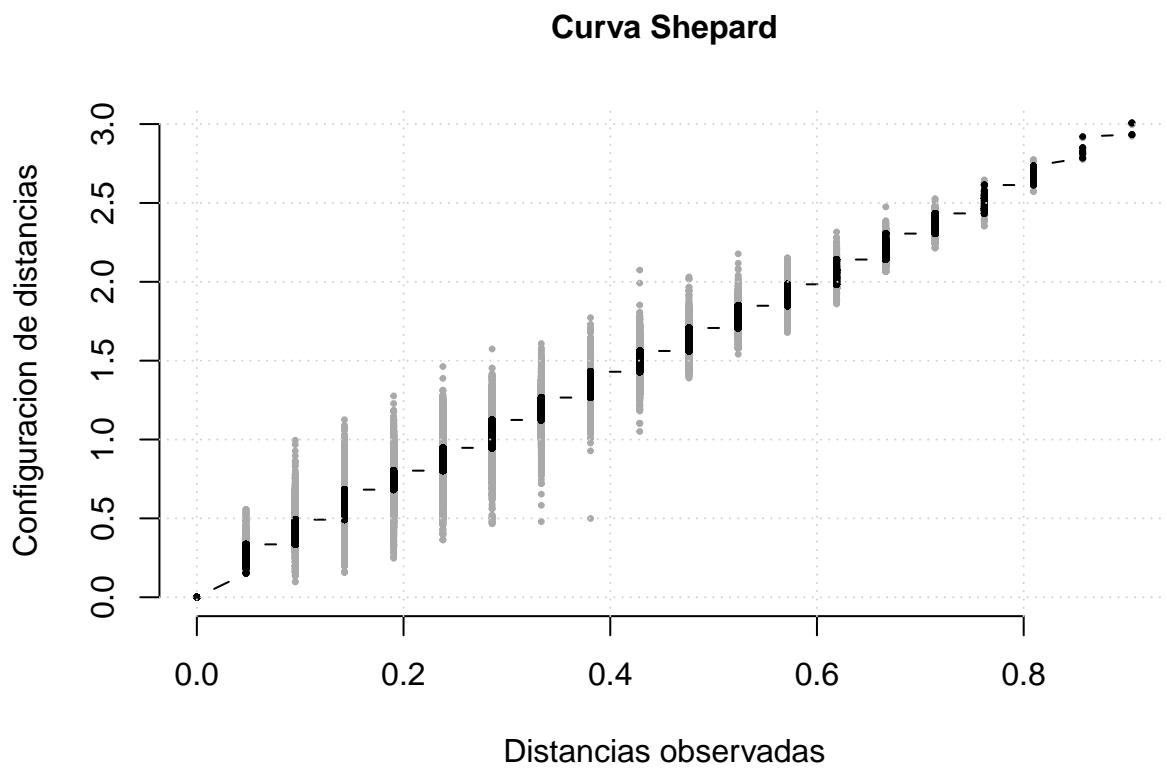
```
## [1] "RSS: 496.003450"
```

```
# Dispersion
plot(fit.datos, plot.dim = c(1,2), main = "Escalamiento Multidimensional No metrico",
      xlab="Dim 1", ylab="Dim 2", cex=0.5, cex.main=1,
      bty = "n", col = datos$Grado.de.rezago.social );grid()
```

Escalamiento Multidimensional No metrico



```
# Curva Shape
plot(fit.datos, plot.type = "Shepard", main="Curva Shepard",
      xlab="Distancias observadas", ylab="Configuracion de distancias", cex=0.5, cex.main=1,
      col="skyblue", bty = "n");grid()
```



Mejoras

Para mejorar el ajuste, se puede intentar los siguiente:

1. Incrementar el numero de dimensiones(Capturar mayor variabilidad que implica menor rss)
2. Usar otra medida de disimilaridad
3. Usar otro algoritmo de optimizacion para el escalamiento
4. Problemas de preprocesamiento
5. Usar otro metodo como t-sne

Implementación de t-sne

```
from sklearn.manifold import TSNE
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Implementacion

```

# Datos
datos = r.datos

# Particion horizontal
x = np.array(datos[r.analisis])
y = np.array(datos[r.auxiliares[1]]) # Variable suplementaria

```

Ajuste

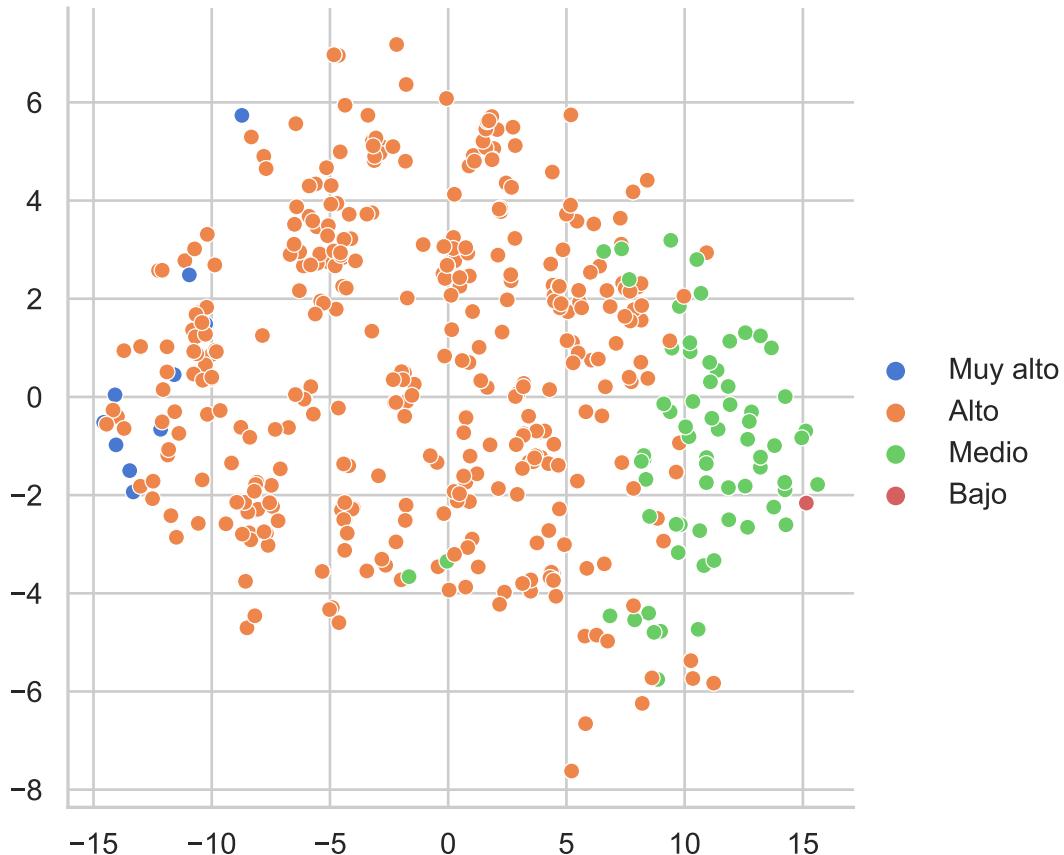
```
x_coord = TSNE(n_components = 3, perplexity = 30, n_iter = 4000).fit_transform(x)
```

Grafico

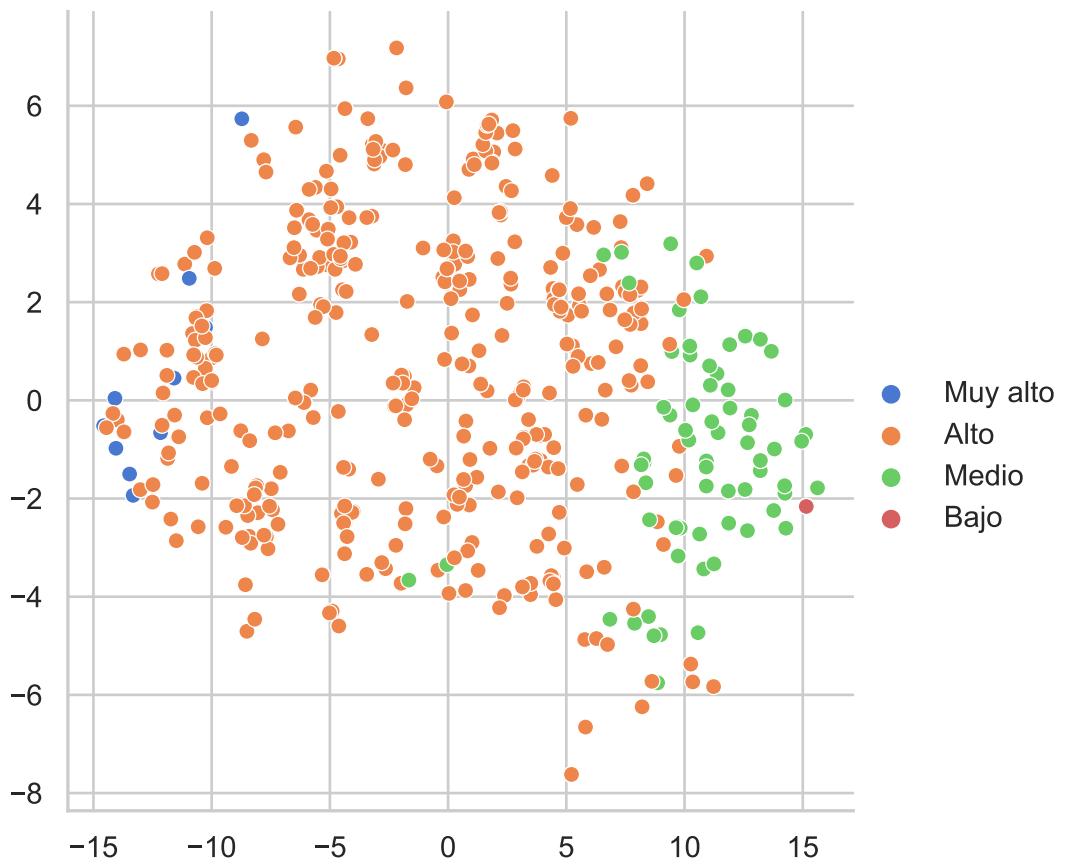
```

plt.clf()
sns.set(style="whitegrid")
sns.relplot(x=x_coord[:,0], y=x_coord[:,1], hue=y, palette="muted" )

```



```
plt.show()
```



```
exit  
## Use exit() or Ctrl-Z plus Return to exit
```

