

## ALGUNOS GRAFICOS PARA REPRESENTAR DATOS

$x_1, \dots, x_n$ , donde  $x_i = (x_{i1}, \dots, x_{ip})$  y  $p > 3$

Idea: Transformer cada  $x_i \in \mathbb{R}^p$  en un objeto que esta en dos dimensiones (ó tres dimensiones).

### 1 LAS CARAS DE CHERNOFF-FLURY

Estas representaciones gráficas, condensan la información de  $x_i$  en una dibujo de una cara.

El tamaño y color de los rasgos en una cara, como lo son: pupilas, ojos, nariz, etc., se asignan dados los valores de  $x_{i1}, x_{i2}, \dots, x_{ip}$ . En algunos paquetes como R, se pueden usar los siguientes rasgos en el dibujo de la cara

- 1 Tamaño del ojo derecho.
- 2 Tamaño de la pupila en el ojo derecho.
- 3 Posición de la pupila en el ojo derecho.
- 4 Inclinação en el ojo derecho.
- 5 Posición horizontal del ojo derecho.
- 6 Posición vertical del ojo derecho.

- 7 curvatura de la ceja derecha.
- 8 densidad de la ceja derecha.
- 9 posición horizontal de la ceja derecha.
- 10 Posición vertical de la ceja derecha.
- 11 Línea superior del cabello (derecha).
- 12 Línea inferior del cabello (derecha).
- 13 Línea derecha de la cara
- 14 Tono del cabello (derecha).
- 15 Inclinación del cabello (derecha)
- 16 Línea de la nariz (derecha)
- 17 Tamaño de la boca (derecha)
- 18 Curvatura de la boca (derecha)

Rasgos del 19 al 36, son los mismos que 1-18 pero para el lado izquierdo.

Cada variable que se asignará a algún rasgo de la cara se transforma primero al intervalo (escala)  $(0,1)$ , de forma que el mínimo



de los posibles valores de esa variable corresponde a 0 y el máximo de los posibles valores corresponde a 1.<sup>(1)</sup> Si por ejemplo el rasgo de la cara al cual fue asignada la variable es el tono del cabello, el individuo de la muestra cuyo valor en la variable sea el de mayor magnitud, podría aparecer en los gráficos como la cara con el tono de cabello más oscuro.

Como ejemplo, vamos a considerar los billetes  $x_{q1}, x_{q2}, \dots, x_{q10}$  en el caso del banco suizo.

Los primeros diez individuos corresponden a billetes verdaderos y los segundos diez corresponden a billetes falsos (de acuerdo a la información que recibimos para hacer la figura E).

(1) Supóngase que se selecciona la variable  $j$   $j \in \{1, 2, \dots, P\}$  entonces  $x_0 \equiv \min \{x_{1j}, x_{2j}, \dots, x_{nj}\}$   
 $x_1 \equiv \max \{x_{1j}, x_{2j}, \dots, x_{nj}\}$ , la transformación lleva  $(x_0, x_1)$  a  $(0, 1)$ .

Para estos datos, vamos a asignar las componentes  $X_1, X_2, \dots, X_6$  a los siguientes rasgos en las caras

$X_1 \rightarrow 1, 19$  tamaños de los ojos.

$X_2 \rightarrow 2, 20$  tamaños de las pupilas.

$X_3 \rightarrow 4, 22$  inclinaciones de los ojos.

$X_4 \rightarrow 11, 29$  líneas superiores del cabello.

$X_5 \rightarrow 12, 30$  líneas inferiores del cabello.

$X_6 \rightarrow 13, 14, 31, 32$  líneas de la cara y tono del cabello

usando estas especificaciones y la función de R "MVAfacebank10.R" obtenemos la figura I

Las caras correspondientes a los dos últimos renglones en la figura I provienen de billetes falsos. Es claro que los dos primeros renglones tienen caras con expresión más "feliz" que las caras de los últimos dos renglones.

```

# -----
# Book:      MVA3
# -----
# See also:   MVAfacebank50
# -----
# Quantnet:   MVAfacebank10
# -----
# Description: MVAfacebank10 computes Flury faces for
#              the Swiss bank notes data ("bank2.dat").
#              Interesting plot is obtained for obs 91-110.
# -----
# Author:     2006-09-15 Julia Wandke
# -----

rm(list=ls(all=TRUE))
graphics.off()

#install.packages("aplpack")
library(aplpack)

# Load data
# The data file should be located in the same folder as this Qlet
# Set the R working directory to this directory using setwd()
# setwd("C:/...")      # set working directory if windows
# setwd("/Users/...")  # set working directory if mac
x = read.table("SwissBank 1.txt")
xx = x[91:110,]

ncolors=15

faces(xx, nrow = 4, face.type=1, scale=TRUE, col.nose = rainbow(ncolors), col.eyes = rainbow(ncolors,
  start = 0.6, end = 0.85), col.hair = terrain.colors(ncolors),
  col.face = heat.colors(ncolors), col.lips = rainbow(ncolors,
  start = 0, end = 1), col.ears = rainbow(ncolors, start = 0,
  end = 0.8), plot.faces = TRUE)

```

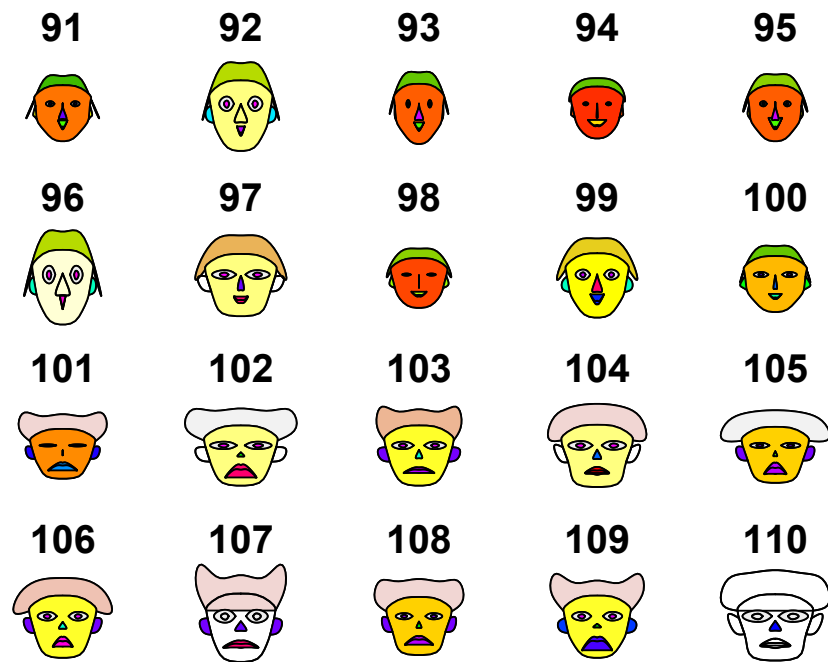


Figura I