PCA para visualización de datos

Victor Gallego y Roi Naveiro

01/04/2019

Funciones auxiliares

- show_digit: Hace una gráfica del dígito en cuestión.
- load_image_file: Para cargar las imágenes de los dígitos
- load_label_file: Para cargar las etiquetas

```
show_digit = function(arr784, col = gray(12:1 / 12), ...) {
  image(matrix(as.matrix(arr784[-785]), nrow = 28)[, 28:1], col = col, ...)
}

load_image_file = function(filename) {
  ret = list()
  f = file(filename, 'rb')
  readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  n    = readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  nrow = readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  ncol = readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  x = readBin(f, 'integer', n = n * nrow * ncol, size = 1, signed = FALSE)
  close(f)
  data.frame(matrix(x, ncol = nrow * ncol, byrow = TRUE))
}

load_label_file = function(filename) {
  f = file(filename, 'rb')
  readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  n = readBin(f, 'integer', n = 1, size = 4, endian = 'big')
  y = readBin(f, 'integer', n = n, size = 1, signed = FALSE)
  close(f)
  y
}
```

Lectura de Datos

Cargamos el dataset MNIST.

```
test  = load_image_file("../../src/t10k-images.idx3-ubyte")
test$y  = as.factor(load_label_file("../../src/t10k-labels.idx1-ubyte"))
```

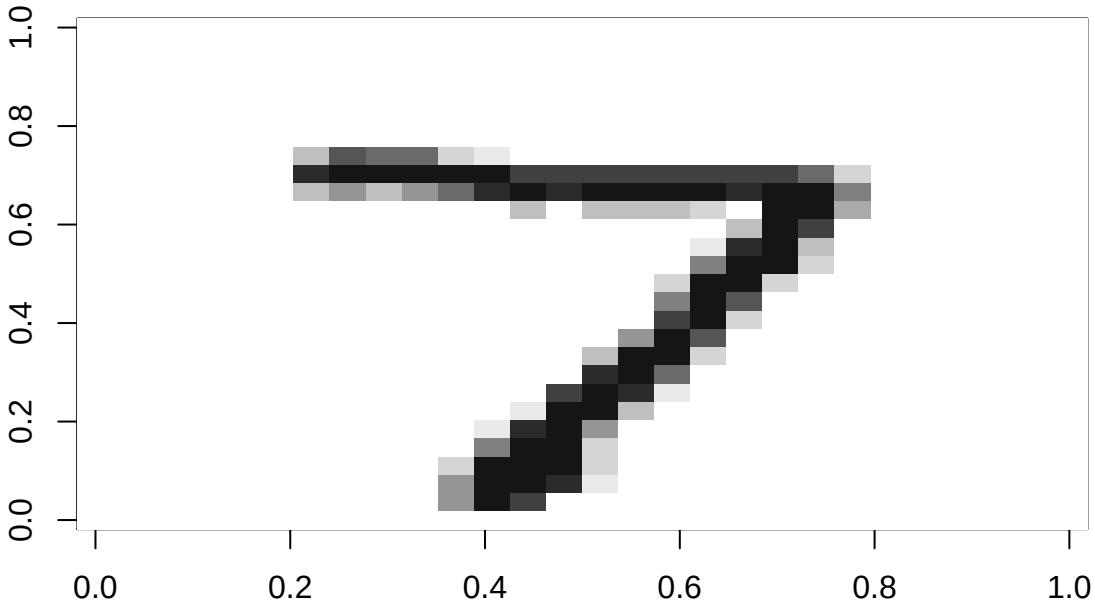
Esta base de datos consta de 10000 imágenes en escala de gris a 28 x 28, de los dígitos del 0 al 9 (escritos a mano).

```
dim(test)
```

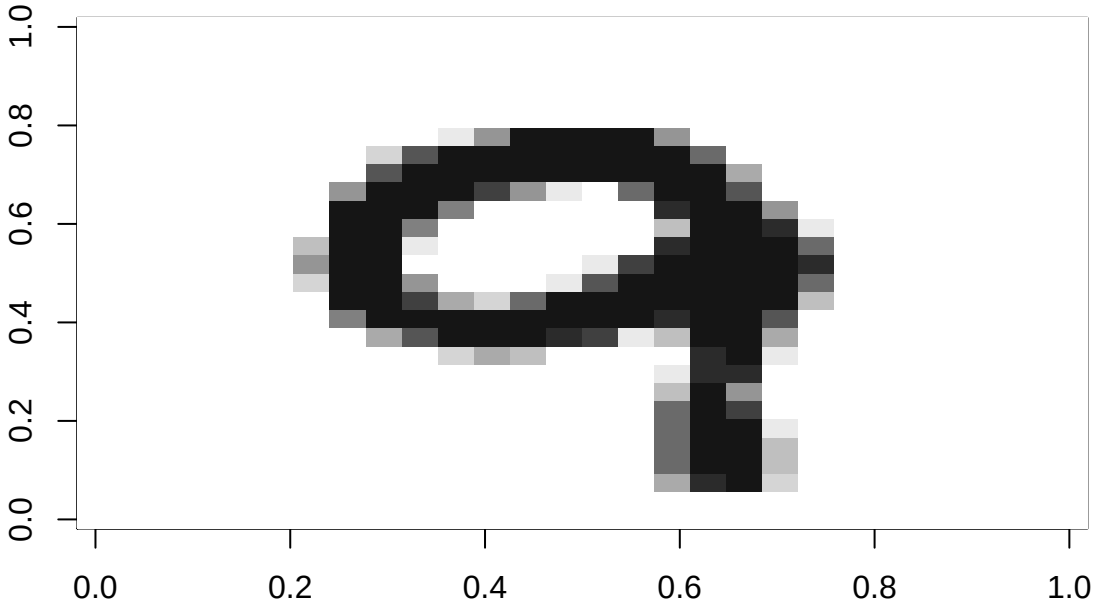
```
## [1] 10000   785
```

Visualizamos algunos ejemplos

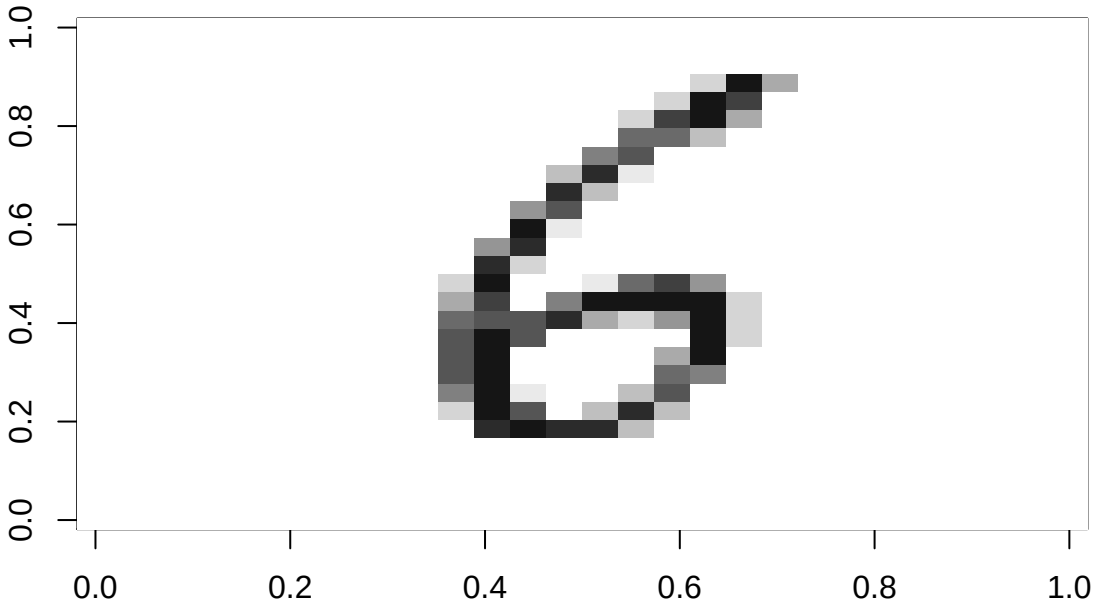
```
show_digit(test[1, ])
```




```
show_digit(test[100, ])
```



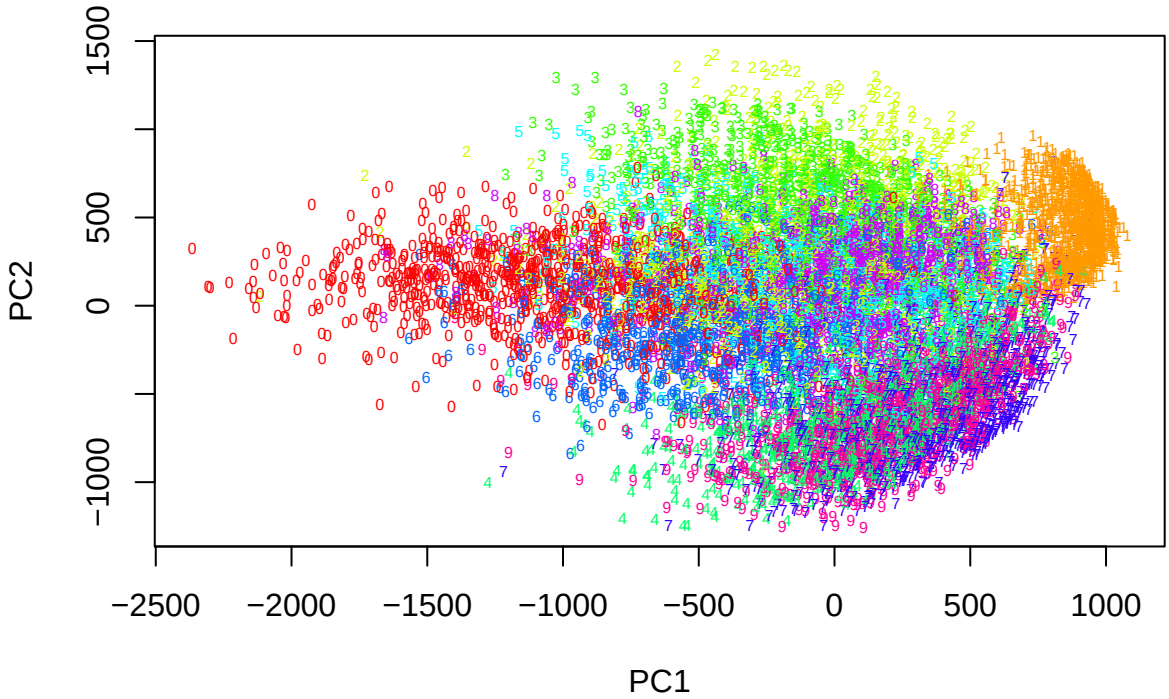

```
show_digit(test[500, ])
```



Proyección a 2D usando PCA

Usaremos el paquete prcomp

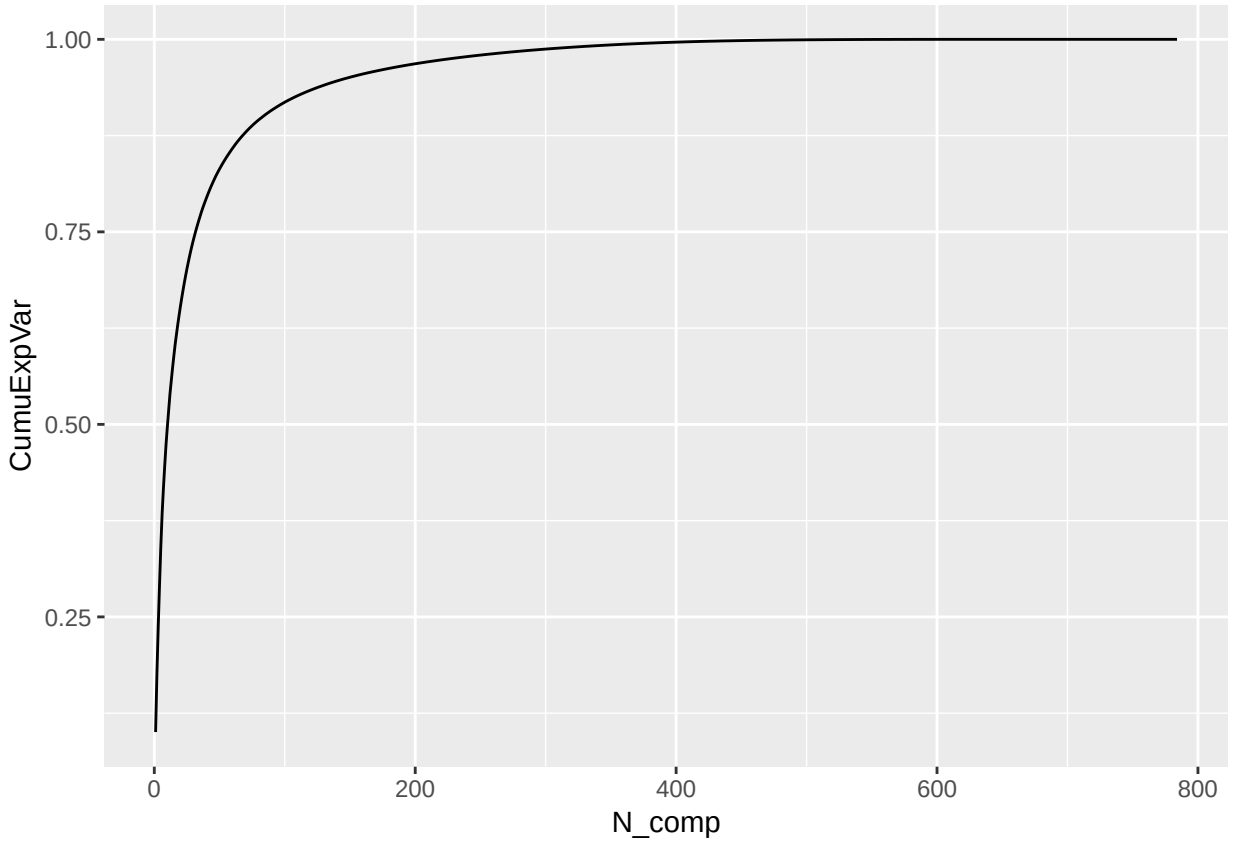
```
proy_pca <- prcomp(test[, 1:28^2], retx = T) ## Ojo, quitar LABEL, sino son trampas
# Representamos las dos primeras componentes
plot(proy_pca$x[, 1:2], type = 'n')
text(proy_pca$x[, 1:2], labels = test$y, cex = 0.5,
     col = rainbow(length(levels(test$y)))[test$y])
```



Pregunta

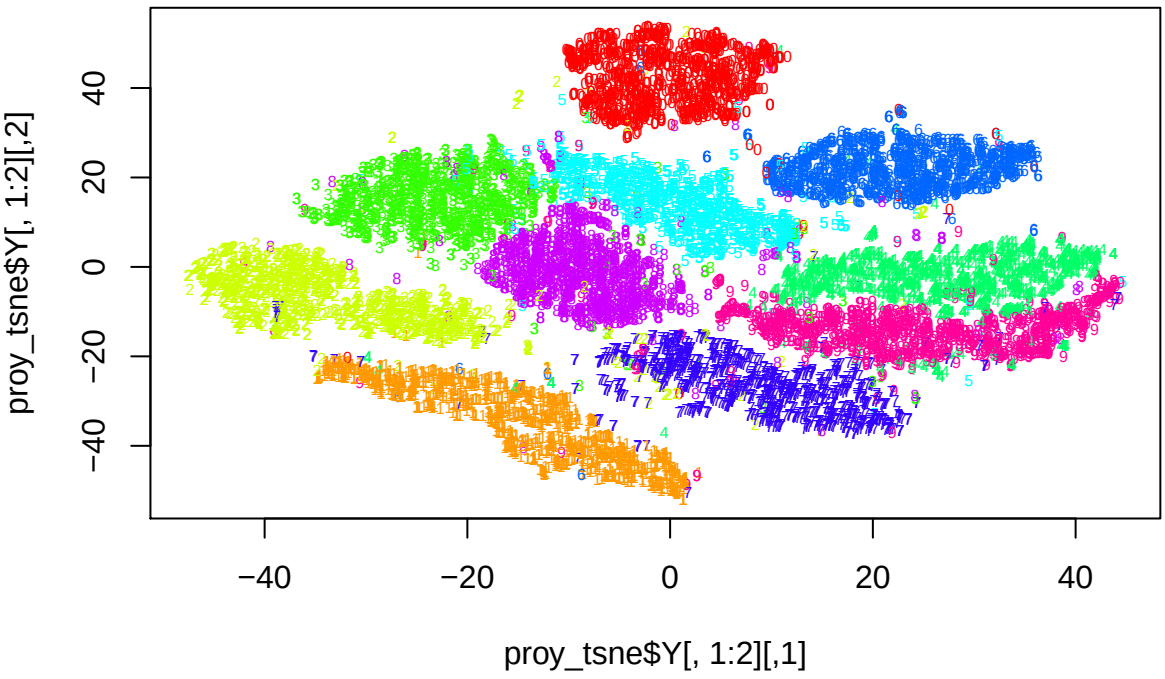
Pinta la curva de número de componentes frente a proporción de varianza explicada varianza explicada. ¿Cuántas componentes son necesarias para explicar el 99% de la varianza?

```
library(ggplot2)
eigs = summary(proy_pca)$sdev^2
#
max_ncom = dim(test)[2]-1
df = data.frame( 1:max_ncom, eigs, eigs/sum(eigs), cumsum(eigs)/sum(eigs))
colnames(df) = c("N_comp", "Eigenvalues", "ExpVar", "CumuExpVar")
p = ggplot(data=df) + geom_line(aes(x = N_comp, y = CumuExpVar) )
p
```



Proyección a 2D usando t-SNE

```
library(Rtsne)
proy_tsne <- Rtsne(test[, 1:28^2], num_threads = 16)
plot(proy_tsne$Y[, 1:2], type = 'n')
text(proy_tsne$Y[, 1:2], labels = test$y, cex = 0.5,
     col = rainbow(length(levels(test$y)))[test$y])
```


proy_tsne$Y[, 1:2][,2]
40
20
0
−20
−40
−40
−20
0
20
40
proy_tsne$Y[, 1:2][,1]