Anàlisi de components principals

Vicent Caselles Ballester

2024-04-07

Exercici 1

a) Escribir la función de densidad $f(x1, x2)$ del vector x y representarla en tres dimensiones.

La primera part feta a mà. No ho mostro, però m'he basat en: https://www.ime.unicamp.br/~cnaber/mvnprop.pdf.

Segona part:

First I'm gonna create a function that converts a covariance matrix into a correlation matrix:

```
cov2cor_vc <- function(cov){
  # from:
  # https://math.stackexchange.com/questions/186959/correlation-matrix-from-covariance-matrix
  D <- diag(sqrt(diag(cov)))
  cor <- solve(D) %*% cov %*% solve(D)
  cor
}
```

Let's try it out:

```
cov1 <- matrix(c(8,5,5,4), ncol=2)
cov2cor_vc(cov1)
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.8838835
## [2,] 0.8838835 1.0000000
```

```
# We see that if we use the "built-in" function from R we get the same
cov2cor(cov1)
```

```
##           [,1]      [,2]
## [1,] 1.0000000 0.8838835
## [2,] 0.8838835 1.0000000
```

Cool.

Bivariate density function given x1, x2, cov matrix.

```
bivariate_df_from_cov <- function(x1, x2, cov){
  # assumes mu = 0
  cor <- cov2cor_vc(cov)
  p12 <- cor[1,2]
  x1_stuff <- x1/(sqrt(cov[1,1]))
  x2_stuff <- x2/sqrt(cov[2,2])
  exp_stuff <- (1/(1-p12**2)) * ((x1_stuff**2) + (x2_stuff**2) - 2*p12*x1_stuff*x2_stuff)
```

```
  exp_stuff <- -exp_stuff/2

  pre_exp <- 1/(2*pi*sqrt(det(cov)))
  fx1x2 <- pre_exp * exp(exp_stuff)
  fx1x2
}
```

Define variables:

```
x1 <- seq(-10, 10, length=41)
x2 <- x1
z_vc <- outer(x1,x2,FUN = bivariate_df_from_cov, cov1) # calculating the density values
```

Plot density function:

```
# Commented because I cannot plot it on pdf.
# library(plotly)
# plot_ly() %>% add_surface(x = x1, y = x2, z = t(z_vc))
```

Now I'm just checking I've gotten the z values correctly:

```
mu1<-0 # setting the expected value of x1
mu2<-0 # setting the expected value of x2
s11 <- 8 # setting the variance of x1
s12 <- 5 # setting the covariance between x1 and x2
s22 <- 4 # setting the variance of x2
rho <- 5/sqrt(8*4) # setting the correlation coefficient between x1 and x2
x1 <- seq(-10, 10, length=41) # generating the vector series x1
x2 <- x1 # copying x1 to x2

f<-function(x1,x2){
term1 <- 1/(2*pi*sqrt(s11*s22*(1-rho^2)))
term2 <- -1/(2*(1-rho^2))
term3 <- (x1-mu1)^2/s11
term4 <- (x2-mu2)^2/s22
term5 <- -2*rho*((x1-mu1)*(x2-mu2))/(sqrt(s11)*sqrt(s22))
term1*exp(term2*(term3+term4+term5))
} # setting up the function of the multivariate normal density >#
z <- outer(x1,x2,f) # calculating the density values

all.equal(z, z_vc)
```

```
## [1] TRUE
```

Cool

b) Realizar un análisis de componentes principales de x.

COV is a 2x2 matrix symmetric matrix, therefore it's eigenvectors form an orthogonal matrix. I has two real eigenvalues.

```
V <- eigen(cov1)$vectors; D <- eigen(cov1)$values
a1 <- V[,1]
a2 <- V[,2]

t(a1)%*%cov1%*%a1
```

```
##          [,1]
```

```
## [1,] 11.38516
```

```
D[1]
```

```
## [1] 11.38516
```

```
t(a2)%*%cov1%*%a2
```

```
##           [,1]
## [1,] 0.6148352
```

```
D[2]
```

```
## [1] 0.6148352
```

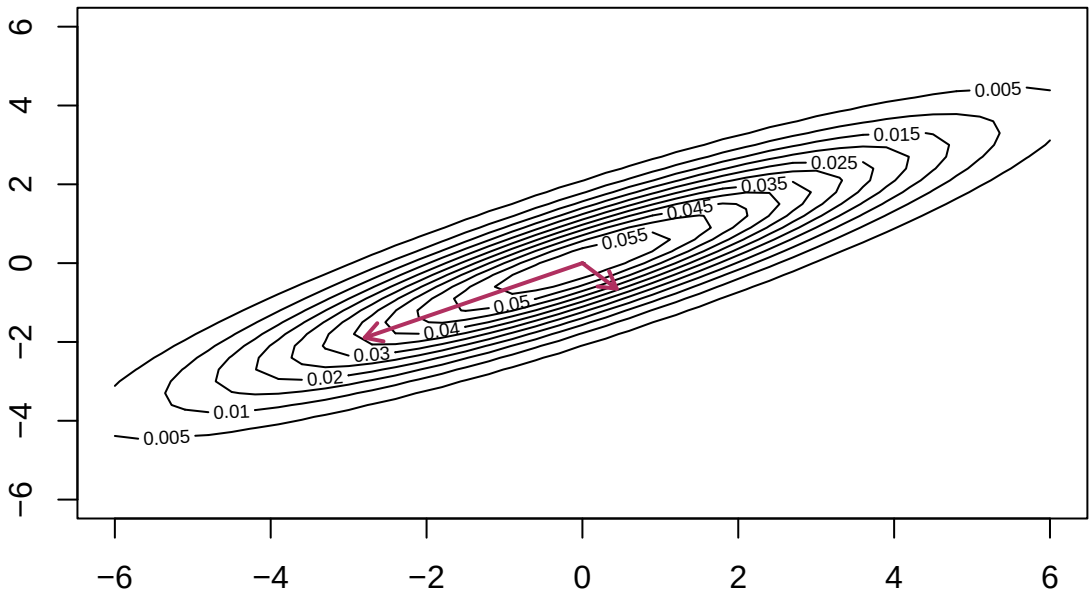
Veiem que la primera component explicaria un % elevat de la variança de les dades originals. Podríem reduir de $p = 2$ a $m = 1$ variables.

c) Dibujar un gráfico de curvas de nivel de la función de densidad en el cuadrado [-6, 6] × [-6, 6] con la función `contour(x,y,z)` de R. Añadir a este gráfico los vectores de las componentes principales con la función `arrows()` y explicar el resultado.

Very much copied from the solution...

Question: Why are the vectors scaled by $\sqrt{\lambda}$?

```
x1 <- seq(-6, 6, length=41); x2 <- x1
z <- outer(x1, x2, FUN=bivariate_df_from_cov, cov1)

contour(x1,x2,z, nlevels=20)
arrows(0, 0,
       a1[1]*sqrt(D[1]),
       a1[2]*sqrt(D[1]),
       len=0.1,lwd=2,col="maroon")

arrows(0,0,
       a2[1]*sqrt(D[2]),
       a2[2]*sqrt(D[2]),
       len=0.1,lwd=2,col="maroon")
```



Clearly the eigenvectors of Σ are the shortest and longest radius of the ellipse.

TODO: Think about how the directions of maximum variance relate to the axis of the pdf of a multivariate normal distribution. Some resources:

- https://fkorona.github.io/ATML/2017_2/Lecture_notes/03C_Normal.pdf
- https://www.cs.princeton.edu/courses/archive/fall10/cos513/notes/2010-11-15.pdf
- https://www.cs.columbia.edu/~djhsu/coms4771-f20/lectures/06-multivariate_gaussians_and_pca.pdf

Exercici 2

```
cov1 <- matrix(c(3,1,1,
                 1,3,1,
                 1,1,5), ncol=3)
```

a) Calcular los valores y vectores propios de Σ

```
V <- eigen(cov1)$vectors
D <- eigen(cov1)$values
a1 <- V[,1]; a2 <- V[,2]; a3 <- V[,3]
t(a1)%*%cov1%*%a1
```

```
##      [,1]
## [1,]    6
```

```
t(a2)%*%cov1%*%a2
```

```
##      [,1]
## [1,]    3
```

```
t(a3)%*%cov1%*%a3
```

```
##      [,1]
## [1,]    2
```

Ho he solucionat a mà, inspirat per la solució. Bàsicament, els eigenvalues que surten són $\lambda_1 = 6$, $\lambda_2 = 3$ i $\lambda_3 = 2$. Aquestes són les solucions a l'equació següent:

$$-\lambda^3 + 11\lambda^2 - 36\lambda + 36 = 0$$

```
eigensolucion <- function(lambda){
  -(lambda**3) + 11*(lambda**2) - 36*lambda + 36
}

eigensolucion(6)
```

```
## [1] 0
```

```
eigensolucion(3)
```

```
## [1] 0
```

```
eigensolucion(2)
```

```
## [1] 0
```

I els eigenvectors són els vectors v_i tal que:

$$\Sigma v_i = \lambda v_i$$

First eigenvector:

```
v1 <- matrix(c(1,1,2), ncol=1)
cov1 %*%v1
```

```
##      [,1]
## [1,]    6
## [2,]    6
## [3,]   12
```

```
6 * v1
```

```
##      [,1]
## [1,]    6
## [2,]    6
## [3,]   12
```

Second eigenvector:

```
v2 <- c(1,1,-1)
cov1 %*% v2
```

```
##      [,1]
## [1,]    3
## [2,]    3
## [3,]   -3
```

```
3 * v2
```

```
## [1]  3  3 -3
```

Third eigenvector:

```
v3 <- c(1,-1,0)
cov1 %*% v3
```

```
##      [,1]
## [1,]    2
```

```
## [2,]   -2
## [3,]    0
```

```
2 * v3
```

```
## [1]  2 -2  0
```

Cool.

b) Escribir el vector $y = (Y_1, Y_2, Y_3)'$ de componentes principales e indicar la proporción de la varianza total que explica cada componente.

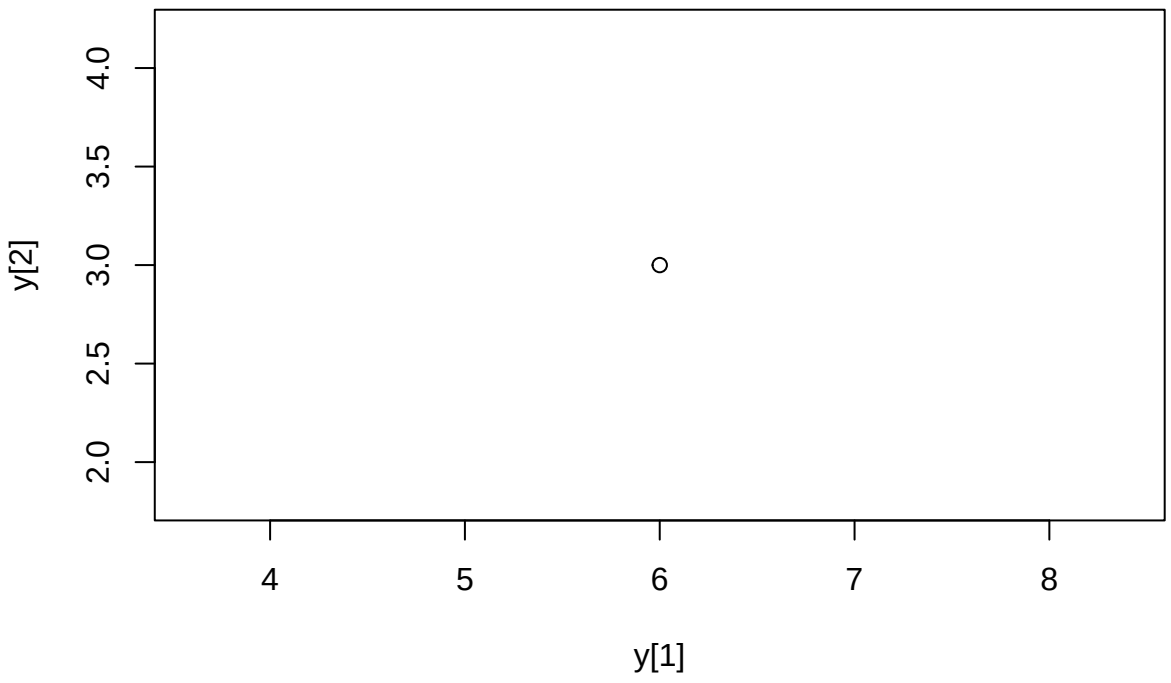
Done a mano. No mostrado.

c) Representar la observación $x = (2, 2, 1)'$ en el plano que definen las dos primeras componentes principales.

```
first_comp <- function(x1, x2, x3)
  x1 + x2 + 2*x3

second_comp <- function(x1, x2, x3)
  x1 + x2 - x3

third_comp <- function(x1,x2,x3) x1 - x2
```

Si queremos solamente las dos primeras componentes:

```
x1 <- 2; x2 <- 2; x3 <- 1
y <- c(first_comp(x1,x2,x3), second_comp(x1,x2,x3))
plot(y[1], y[2])
```



Exercici 3

a) Realizar un análisis de componentes principales y calcular la proporción de varianza explicada por las tres primeras componentes.

Loading the data.

```
load('gorriones.RData')
colnames(gorriones) <- c("length","wing","head","humerus","sternum","survival")
```

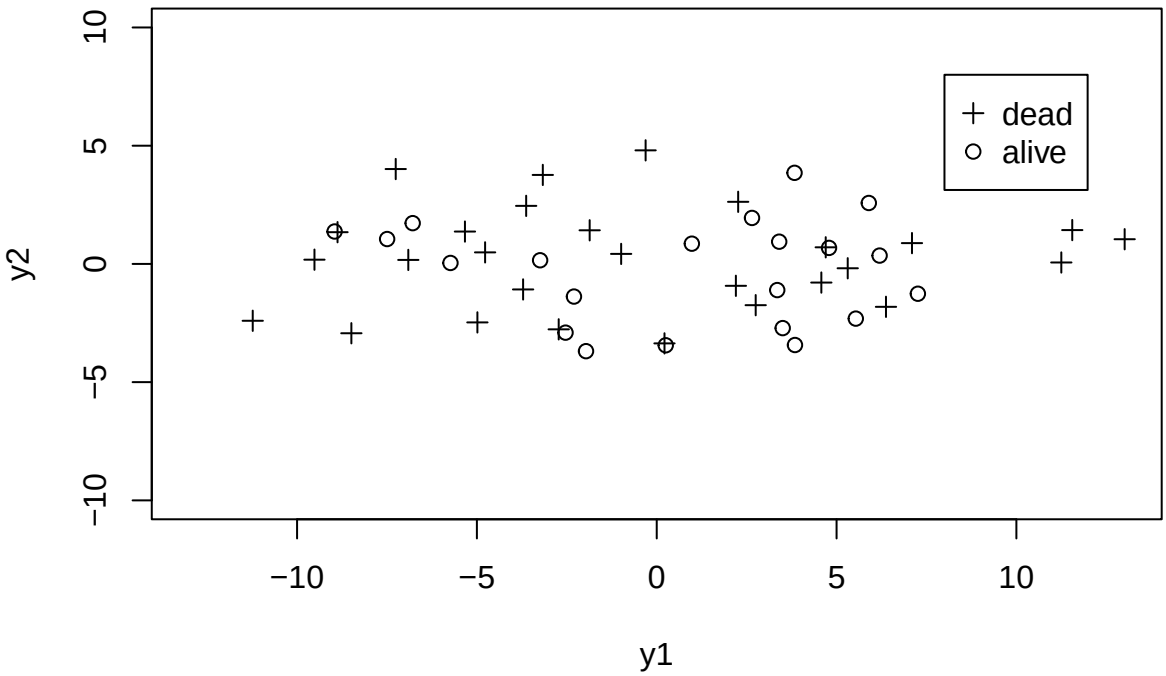
We define the matrix X with the data (without `survival`).

```
X <- data.matrix(gorriones[, -6])
X_scaled = scale(X, scale=F)
S <- 1/(nrow(X)-1) * (t(X_scaled) %*%X_scaled)

V <- eigen(S)$vectors
D <- eigen(S)$values

prvar <- D/sum(D)
round(prvar * 100, 2)
```

```
## [1] 86.22 11.28  1.54  0.76  0.19
```

We see that with the first two principal components we already can explain > 90 of variance from the original data.

```
y1 <- X_scaled%*%V[,1]
y2 <- X_scaled%*%V[,2]
labs <- as.numeric(gorriones$survival)
plot(y1, y2, pch=c(3,21)[labs], ylim = c(-10,10), xlim=c(-13,13))
legend(8, 8, pch=c(3,21), c("dead", "alive"))
```



We see that most sparrows that survived are clustered along higher y_1 (first component values). If we inspect the "weights" that construct y_1 from the original variables we see that they are a negative linear combination of

all the original variables, with greater weights on the first two ones, *length* and *wing*.

```
V[,1]
```

```
## [1] -0.53650052 -0.82901535 -0.09649615 -0.07435219 -0.10030441
```

So we can assume that bigger sparrows tend to survive more? Also we could assume that individuals with *extreme size* values are less likely to survive.

Exercici 4

Carreguem les dades.

```
data(crabs, package='MASS')
str(crabs) # les variables "numèriques" que podem utilitzar per a PCA són les 5 últimes
```

```
## 'data.frame':    200 obs. of  8 variables:
##  $ sp   : Factor w/ 2 levels "B","O": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex  : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ index: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ FL   : num  8.1 8.8 9.2 9.6 9.8 10.8 11.1 11.6 11.8 11.8 ...
##  $ RW   : num  6.7 7.7 7.8 7.9 8 9 9.9 9.1 9.6 10.5 ...
##  $ CL   : num  16.1 18.1 19 20.1 20.3 23 23.8 24.5 24.2 25.2 ...
##  $ CW   : num  19 20.8 22.4 23.1 23 26.5 27.1 28.4 27.8 29.3 ...
##  $ BD   : num  7 7.4 7.7 8.2 8.2 9.8 9.8 10.4 9.7 10.3 ...
```

```
X <- crabs[4:8]
X_scaled <- scale(X, scale=F)
S <- 1/(nrow(X)-1) * t(X_scaled)%*%X_scaled
V <- eigen(S)$vectors; D <- eigen(S)$values
prvar <- D/sum(D)
round(prvar, 2)
```

```
## [1] 0.98 0.01 0.01 0.00 0.00
```

Veiem que amb una PC ja cobrim el 98% de la variança de les dades originals.