UNIVERSITAT DE BARCELONA

Universitat Oberta de Catalunya

Análisis de componentes principales
Soluciones

Francesc Carmona y Josep Gregori*

11 de abril de 2023

Ejercicio 1

Sea $\mathbf{x}$ *un vector aleatorio que sigue una distribución normal bivariante de media cero y matriz de covarianzas*

$$\mathbf{\Sigma} = \begin{pmatrix} 8 & 5 \\ 5 & 4 \end{pmatrix}$$

a) *Escribir la función de densidad* $f(x_1, x_2)$ *del vector* $\mathbf{x}$ *y representarla en tres dimensiones*[1].

La función de densidad de una distribución normal bivariante $\mathbf{x}$ es

$$f(x_1, x_2) = \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22}(1-\rho^2)}} \exp\left[-\frac{z}{2(1-\rho^2)}\right]$$

donde

$$z = \frac{(x_1-\mu_1)^2}{\sigma_{11}} - 2\rho\frac{x_1-\mu_1}{\sqrt{\sigma_{11}}}\frac{x_2-\mu_2}{\sqrt{\sigma_{22}}} + \frac{(x_2-\mu_2)^2}{\sigma_{22}}$$

y

$$\rho = \frac{\sigma_{12}}{\sqrt{\sigma_{11}}\sqrt{\sigma_{22}}}$$

En este caso podemos concretar como

$$f(x_1, x_2) = \frac{1}{2\pi\sqrt{8\cdot 4(1-25/32)}} \exp\left[-\frac{z}{2(1-25/32)}\right]$$

y

$$z = \frac{x_1^2}{8} - \frac{50}{32}\frac{x_1}{2\sqrt{2}}\frac{x_2}{2} + \frac{x_2^2}{4}$$

```
> # 3-D plots
> #
> 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
```

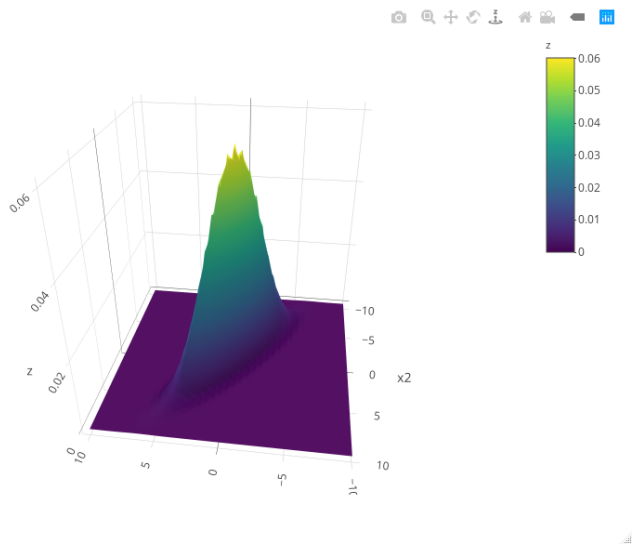
*Alumno del curso 2009-10

[1] El código de la página 2 del documento `http://www.ejwagenmakers.com/misc/Plotting_3d_in_R.pdf` puede ser útil. No confundir la correlación entre dos variables con su covarianza.

```
> #
> 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
```

Ahora podemos hacer un gráfico 3D con ayuda del paquete plotly que se puede ver en un navegador o en RStudio.

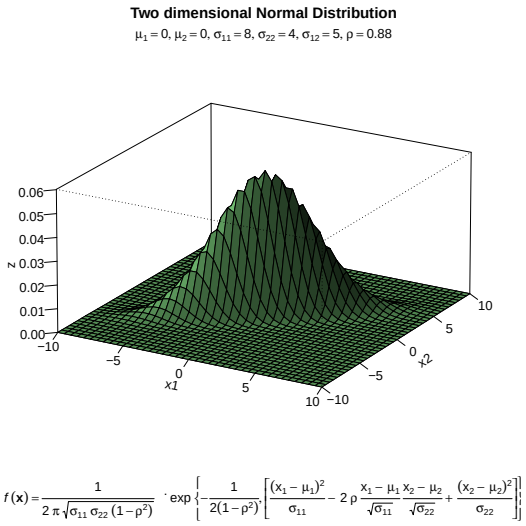
```
> library(plotly)
> plot_ly() %>% add_surface(x = x1, y = x2, z = t(z))
```



O también, con más trabajo, con la función `persp()`.

```
> #
> persp(x1, x2, z,
+ main="Two dimensional Normal Distribution",
+ sub=expression(italic(f)~(bold(x))==frac(1,2~pi~sqrt(sigma[11]~
+ sigma[22]~(1-rho^2)))~phantom(0)^bold(.)~exp~bgroup("{",
+ list(-frac(1,2(1-rho^2)),
+ bgroup("[", frac((x[1]~-~mu[1])^2, sigma[11])~-~2~rho~frac(x[1]~-~mu[1],
+ sqrt(sigma[11]))~ frac(x[2]~-~mu[2],sqrt(sigma[22]))~+~
+ frac((x[2]~-~mu[2])^2, sigma[22]),"]")),"}")),
+ col="lightgreen",
+ theta=30, phi=20,
+ r=50,
+ d=0.1,
```

```
+ expand=0.5,
+ ltheta=90, lphi=180,
+ shade=0.75,
+ ticktype="detailed",
+ nticks=5) # produces the 3-D plot
> #
> mtext(expression(list(mu[1]==0,mu[2]==0,sigma[11]==8,sigma[22]==4,sigma[12]==5,
+                   rho==0.88)), side=3) # adding a text line to the graph
```



$$f(\mathbf{x}) = \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22}(1-\rho^2)}} \cdot \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\frac{(x_1-\mu_1)^2}{\sigma_{11}} - 2\rho\frac{x_1-\mu_1}{\sqrt{\sigma_{11}}}\frac{x_2-\mu_2}{\sqrt{\sigma_{22}}} + \frac{(x_2-\mu_2)^2}{\sigma_{22}}\right]\right\}$$

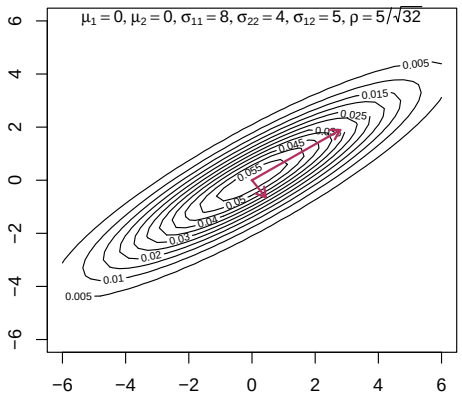
b) *Realizar un análisis de componentes principales de* $\mathbf{x}$.

```
> Sigma <- matrix(c(8,5,5,4), ncol=2)
> evv <- eigen(Sigma)
> evv

eigen() decomposition
$values
[1] 11.3851648  0.6148352

$vectors
           [,1]       [,2]
[1,] -0.8280672  0.5606288
[2,] -0.5606288 -0.8280672
```

c) *Dibujar un gráfico de curvas de nivel de la función de densidad en el cuadrado* $[-6,6]\times[-6,6]$ *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.*

```
> # Gráfico de curvas de nivel
> x1 <- seq(-6, 6, length=41)
> x2 <- seq(-6, 6, length=41)
> z <- outer(x1,x2,f)
> contour(x1,x2,z, nlevels=20)
> mtext(expression(list(mu[1]==0,mu[2]==0,sigma[11]==8,sigma[22]==4,
```

```
+ sigma[12]==5,rho==5/sqrt(32))), side=3, line=-1, cex=0.88)
> arrows(0,0,-evv$ve[1,1]*sqrt(evv$va[1]),-evv$ve[2,1]*sqrt(evv$va[1]),
+ len=0.1,lwd=2,col="maroon")
> arrows(0,0,evv$ve[1,2]*sqrt(evv$va[2]),evv$ve[2,2]*sqrt(evv$va[2]),
+ len=0.1,lwd=2,col="maroon")
```



El primer vector propio está claramente en la dirección de máxima variabilidad, es decir, en el eje mayor de las elipses. Hemos cambiado el signo del vector, ya que lo importante es la dirección y no el sentido.

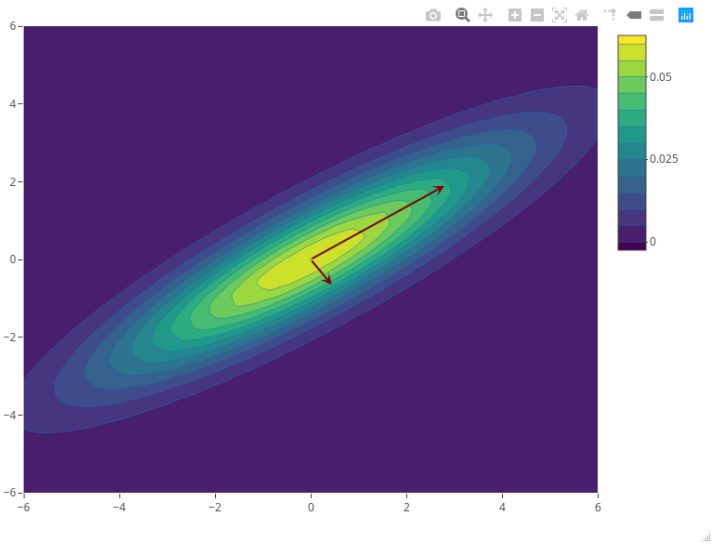
El segundo vector propio es perpendicular al primero y en la dirección de máxima variabilidad, es decir, en el eje secundario de las elipses.

Con el paquete `plotly` se puede hacer un gráfico más vistoso.

```
> p <- plot_ly(x = x1, y = x2, z = t(z), type = "contour")
```

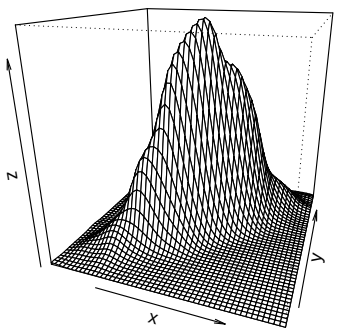
y ahora añadimos los vectores de las componentes principales

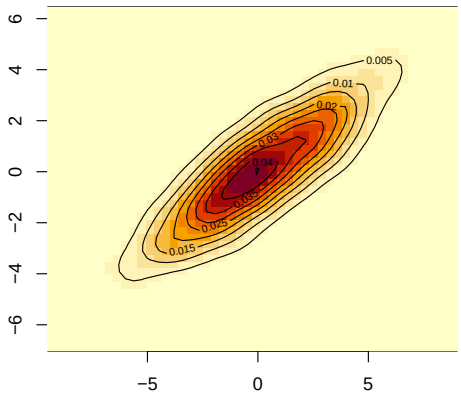
```
> p %>%
+   add_annotations(text="",
+                   x = -evv$ve[1,1]*sqrt(evv$va[1]),
+                   y = -evv$ve[2,1]*sqrt(evv$va[1]),
+                   xref = "x", yref = "y",
+                   showarrow = TRUE, arrowcolor='maroon',
+                   arrowhead = 3, arrowsize = 1,
+                   ax = 0, ay = 0,
+                   axref="x", ayref="y") %>%
+   add_annotations(text="",
+                   x = evv$ve[1,2]*sqrt(evv$va[2]),
+                   y = evv$ve[2,2]*sqrt(evv$va[2]),
+                   xref = "x", yref = "y",
+                   showarrow = TRUE, arrowcolor='maroon',
+                   arrowhead = 3, arrowsize = 1,
+                   ax = 0, ay = 0,
+                   axref="x", ayref="y")
```



(∗) *Una alternativa gráfica menos exacta para resolver este ejercicio sería simular unos datos aleatorios y utilizar una estimación de la densidad con el siguiente código:*

```
> # lets first simulate a bivariate normal sample
> library(MASS)
> bivn <- mvrnorm(1000, mu = c(0, 0), Sigma = matrix(c(8, 5, 5, 4), 2))
> # now we do a kernel density estimate
> bivn.kde <- kde2d(bivn[,1], bivn[,2], n = 50)
> # perspective
> persp(bivn.kde, xlab="x", ylab="y", zlab="z", phi = 20, theta = 20)
> # fancy contour with image
> image(bivn.kde); contour(bivn.kde, add = T)
```





Ejercicio 2

Sea la matriz de varianzas-covarianzas poblacionales

$$\mathbf{\Sigma} = \begin{pmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 5 \end{pmatrix}$$

correspondiente a un vector aleatorio $\mathbf{x} = (X_1, X_2, X_3)'$ *de media cero.*

a) *Calcular los valores y vectores propios de* $\mathbf{\Sigma}$.

```
> Sigma <- matrix(c(3,1,1,
+                   1,3,1,
+                   1,1,5), ncol=3, byrow=T)
> evv <- eigen(Sigma)
> evv

eigen() decomposition
$values
[1] 6 3 2

$vectors
           [,1]       [,2]          [,3]
[1,] -0.4082483 -0.5773503  7.071068e-01
[2,] -0.4082483 -0.5773503 -7.071068e-01
[3,] -0.8164966  0.5773503 -1.110223e-16
```

Si solucionamos "a mano" el sistema homogéneo $\mathbf{\Sigma} - \lambda\mathbf{I} = \mathbf{0}$ para $\lambda = 6$, la solución indeterminada es $x(1, 1, 2)'$ de modo que podemos tomar $(1, 1, 2)'$ como primer vector propio de $\mathbf{\Sigma}$. Observar que el vector propio que da **R** es proporcional a éste.

Del mismo modo para el valor propio $\lambda = 3$, el vector propio asociado puede ser $(1, 1, -1)'$ y para el valor propio $\lambda = 2$, el vector propio es $(1, -1, 0)'$.

Los vectores propios que da **R** están normalizados y, en este caso, no son tan elegantes como los que se pueden calcular a mano.

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

Las componentes principales son

$$\begin{aligned} Y_1 &= X_1 + X_2 + 2X_3 \\ Y_2 &= X_1 + X_2 - X_3 \\ Y_3 &= X_1 - X_2 \end{aligned}$$

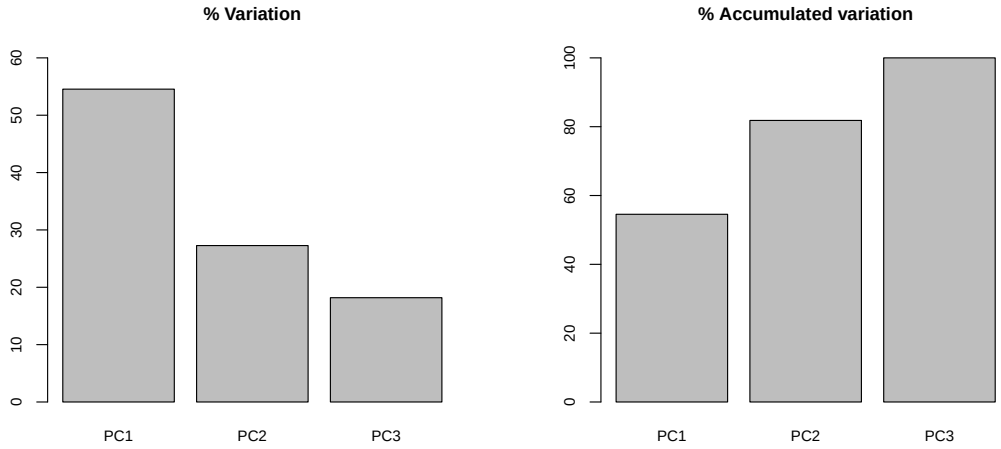
o matricialmente

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 1 & -1 \\ 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

Observemos que los vectores propios son las filas de la matriz de transformación.

La proporción de la varianza total explicada por cada componente:

```
> evals <- evv$values
> names(evals) <- c("PC1","PC2","PC3")
> par(cex.main=1, cex.lab=0.8, cex.axis=0.8)
> barplot(evals/sum(evals)*100,main="% Variation",ylim=c(0,60))
> barplot(cumsum(evals)/sum(evals)*100,ylim=c(0,100),
+         main="% Accumulated variation")
```



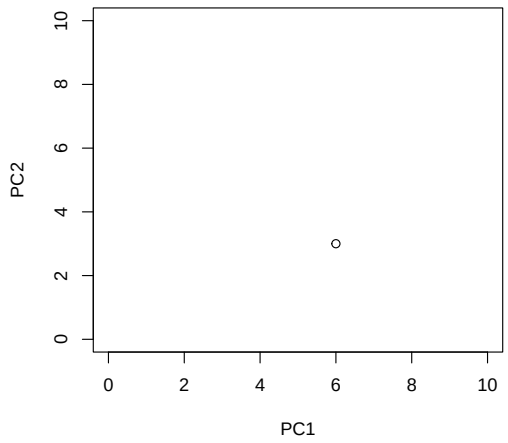
c) *Representar la observación* $\mathbf{x} = (2, 2, 1)'$ *en el plano que definen las dos primeras componentes principales.*

Los *scores* de esta observación son

$$\begin{pmatrix} 1 & 1 & 2 \\ 1 & 1 & -1 \\ 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 3 \\ 0 \end{pmatrix}$$

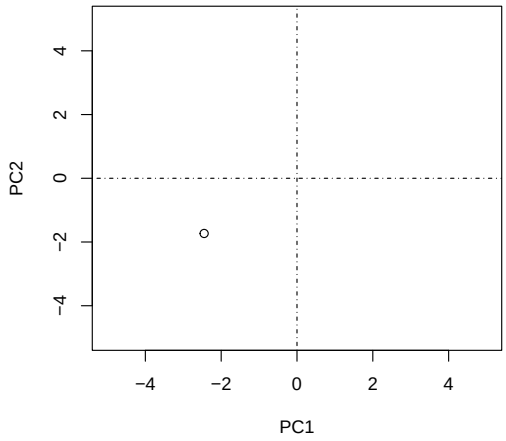
de modo que la representación es

```
> plot(6,3, xlim=c(0,10), ylim=c(0,10), xlab="PC1", ylab="PC2")
```



Con los vectores propios de **R**

```
> scores <- t(evv$vectors) %*% c(2,2,1)
> plot(scores[1],scores[2], xlim=c(-5,5), ylim=c(-5,5), xlab="PC1", ylab="PC2")
> abline(h=0,v=0,lty=4)
```



Ejercicio 3

Con los datos de las $p = 5$ medidas biométricas de los $n = 49$ gorriones hembras que fueron recogidos casi moribundos después de un temporal,

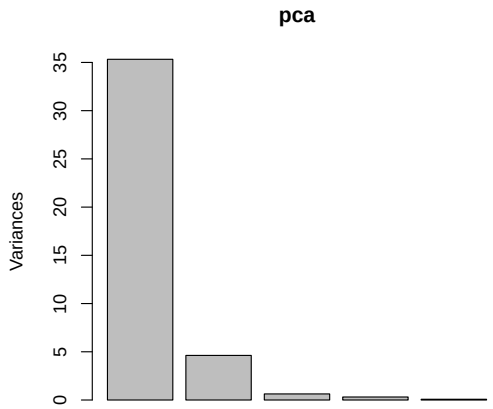
Figura 1: House sparrow (Passer domesticus).

a) *Realizar un análisis de componentes principales*[2] *y calcular la proporción de varianza explicada por las tres primeras componentes.*

```
> load("gorriones.RData")
> str(gorriones)
```

[2]En **R** se utiliza la función `prcomp()`, aunque también existe `princomp()` semejante a la función de S-PLUS. Las funciones `print`, `plot` y `biplot` se aplican sobre un objeto de la clase "prcomp".

```
'data.frame': 49 obs. of  6 variables:
 $ x1      : num  156 154 153 153 155 163 157 155 164 158 ...
 $ x2      : num  245 240 240 236 243 247 238 239 248 238 ...
 $ x3      : num  31.6 30.4 31 30.9 31.5 32 30.9 32.8 32.7 31 ...
 $ x4      : num  18.5 17.9 18.4 17.7 18.6 19 18.4 18.6 19.1 18.8 ...
 $ x5      : num  20.5 19.6 20.6 20.2 20.3 20.9 20.2 21.2 21.1 22 ...
 $ superviv: Factor w/ 2 levels "N","S": 2 2 2 2 2 2 2 2 2 2 ...

> colnames(gorriones) <- c("length","wing","head","humerus","sternum","survival")
> pca <- prcomp(gorriones[,1:5])
> summary(pca)

Importance of components:
                          PC1    PC2    PC3     PC4     PC5
Standard deviation     5.9435 2.1500 0.7943 0.55927 0.27843
Proportion of Variance 0.8622 0.1128 0.0154 0.00763 0.00189
Cumulative Proportion  0.8622 0.9751 0.9905 0.99811 1.00000

> plot(pca)
```



La proporción de varianza explicada por las tres primeras componentes es de 99.0 % y con las dos primeras ya tenemos un 97.5 %.

b) *Representar los datos sobre el plano de las dos primeras componentes, con dos símbolos distintos en función de si el gorrión sobrevivió o no, e interpretar los ejes como factores de tamaño (el primero) y forma (el segundo).*

A la vista del gráfico, ¿qué gorriones, según tamaño y forma, tienen menos posibilidades de sobrevivir?

La interpretación de las componentes se puede hacer a la vista de los coeficientes que proporcionan dichas componentes:

```
> pca$rotation[,1:2]

                PC1         PC2
```

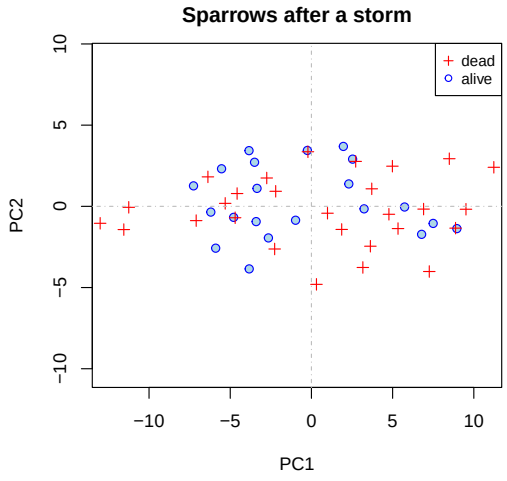
```
length  0.53650052 -0.82809990
wing    0.82901535  0.55051223
head    0.09649615 -0.03356237
humerus 0.07435219  0.01459529
sternum 0.10030441 -0.09923405
```

Los vectores propios quedan indeterminados por el producto por un escalar, en particular, por un factor -1. Así podemos ver como algunos vectores propios obtenidos por diagonalización directa de la matriz de covarianzas de los datos centrados, presentan todos sus signos invertidos respecto a los proporcionados por la función `prcomp()`. Esto repercute en la simetría de la representación de datos según una u otra base. Cuando hay un factor de -1, los datos aparecen reflejados sobre el eje correspondiente.

La primera componente tiene todos los coeficientes positivos (del mismo signo) y se puede interpretar como el tamaño del gorrión. El peso principal de las variables originales se da en las dos primeras. El mayor peso se da a la extensión de las alas, con un coeficiente de 0.83. Le sigue la longitud del pájaro con un coeficiente de 0.54. Las demás variables presentan una contribución notablemente menor a la PC1.

La segunda componente tiene coeficientes positivos y negativos y se puede interpretar como un factor forma. En este caso por contraposición de las dos primeras variables. En el eje PC2 las variables con mayor peso en valor absoluto son las mismas pero aquí una aparece con signo positivo y la otra con signo negativo: 0.55 para las alas, 0.83 para la longitud. Valores positivos de PC2 corresponderán a pájaros cortos con gran envergadura de alas, y valores negativos de PC2 corresponderán a pájaros más proporcionados o incluso de mayor longitud que envergadura.

```
> library(MASS)
> gr <- as.numeric(gorriones$survival)
> eqscplot(pca$x[,1], pca$x[,2], pch=c(3,21)[gr],
+                                col=c("red","blue")[gr],
+                                bg="lightblue",
+                                xlab="PC1",ylab="PC2")
> abline(h=0,v=0,lty=4,col="gray")
> legend("topright",pch=c(3,21),col=c("red","blue"), cex=0.8,
+        legend=c("dead","alive"))
> title(main="Sparrows after a storm",line=1)
```



Es evidente que los gorriones más grandes y los más pequeños que muestra PC1 (por tamaño), no sobrevivieron.

Ejercicio 4

Con los datos de los cangrejos de la base de datos crabs *del paquete* MASS *de* **R**,

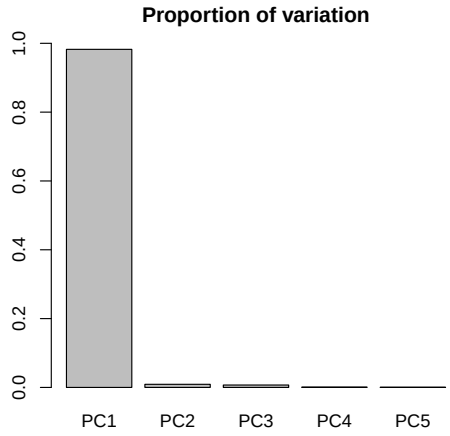
Figura 2: Leptograpsus variegatus.

a) *Realizar un análisis de componentes principales de las cinco medidas biométricas y estudiar la bondad de la representación en dos y en tres dimensiones.*

```
> require(MASS)
> str(crabs)

'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]
> pcx <- prcomp(X); pcx

Standard deviations (1, .., p=5):
[1] 11.8619441  1.1387874  1.0001346  0.3678306  0.2791312

Rotation (n x k) = (5 x 5):
         PC1        PC2        PC3        PC4        PC5
FL 0.2889810  0.3232500 -0.5071698  0.7342907  0.1248816
RW 0.1972824  0.8647159  0.4141356 -0.1483092 -0.1408623
CL 0.5993986 -0.1982263 -0.1753299 -0.1435941 -0.7416656
CW 0.6616550 -0.2879790  0.4913755  0.1256282  0.4712202
BD 0.2837317  0.1598447 -0.5468821 -0.6343657  0.4386868

> summary(pcx)
```

```
Importance of components:
                          PC1     PC2     PC3     PC4     PC5
Standard deviation     11.8619 1.13879 1.00013 0.36783 0.27913
Proportion of Variance  0.9825 0.00906 0.00698 0.00094 0.00054
Cumulative Proportion   0.9825 0.99153 0.99851 0.99946 1.00000
```

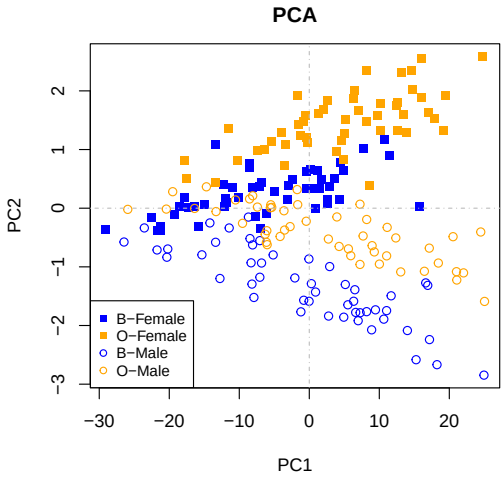
```
> barplot(summary(pcx)$importance[2,],ylim=c(0,1))
> title(main="Proportion of variation",line=1)
```



La diferencia entre utilizar dos o tres componentes es realmente pequeña.

b) *Representar los datos con las dos primeras componentes con dos colores, según la especie, y dos símbolos, segun el sexo. Indicar, si existe, alguna interpretación.*

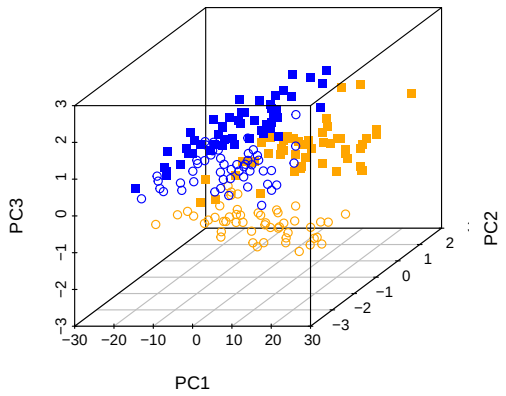
```
> plot(pcx$x[,1:2],col=ifelse(crabs$sp == "O","orange","blue"),
+                  pch=ifelse(crabs$sex == "M", 1, 15),
+                  xlab="PC1",ylab="PC2")
> abline(h=0,v=0,lty=4,col="gray")
> legend("bottomleft",pch=c(15,15,1,1),col=c("blue","orange","blue","orange"),
+ cex=0.8,legend=c("B-Female","O-Female","B-Male","O-Male"))
> title(main="PCA",line=1)
```



Este resultado indica que la mayor variabilidad se retiene con las dos primeras componentes principales. Éstas se ajustan a la clásica interpretación de tamaño, la primera, y forma, la segunda. El tamaño es una combinación ponderada de todas las variables con especial peso de las dos del caparazón. La forma viene definida por la contraposición de las variables del caparazón frente a las otras. En la figura anterior tenemos los datos de las dos primeras componentes con diferentes colores para las especies y diferentes símbolos para el sexo. El gráfico muestra que a mayor tamaño los cangrejos se van diferenciando en la forma según sexo y según especie. La mayor diferencia parece establecerse por sexo, más que por especie.

c) *Representar los datos con las tres primeras componentes.*

```
> library(scatterplot3d)
> scatterplot3d(pcx$x[,1], pcx$x[,2], pcx$x[,3],
+               xlab="PC1", ylab="PC2", zlab="PC3",
+               color = ifelse(crabs$sp == "O", "orange", "blue"),
+               pch = ifelse(crabs$sex == "M", 1, 15))
```



Con el paquete plotly obtenemos un gráfico 3D que se puede mover.

```
> library(plotly)
> fig <- plot_ly(crabs, x = ~pcx$x[,1], y = ~pcx$x[,2], z = ~pcx$x[,3],
+                color = ~sp, colors = c("blue","orange"),
+                symbol = ~sex, symbols = c('circle','x','o')) %>%
+        add_markers(marker = list(size = 3)) %>%
+        layout(scene = list(xaxis = list(title = "PC1"),
+                            yaxis = list(title = "PC2"),
+                            zaxis = list(title = "PC3")))
> fig
```

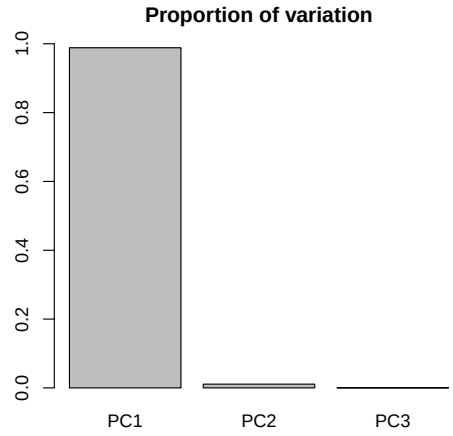
d) *El Análisis de componentes principales puede servir para estudiar la capacidad. Supongamos que el caparazón del cangrejo tiene lognitud L, ancho A y alto H. La capacidad sería $C = L^\alpha A^\beta H^\gamma$, donde α, β y γ son parámetros. Aplicando logaritmos, obtenemos*

$$\log C = \log(L^\alpha A^\beta H^\gamma) = \alpha \log L + \beta \log A + \gamma \log H$$

que podemos interpretar como la primera componente principal de las variables transformadas.

Obtener las capacidades del caparazón de los cangrejos y un gráfico comparativo según especie y sexo.

```
> Xc <- crabs[,c("CL","CW","BD")]
> log.pcx <- prcomp(log(Xc)); log.pcx

Standard deviations (1, .., p=3):
[1] 0.41740432 0.04386825 0.01042033

Rotation (n x k) = (3 x 3):
         PC1        PC2        PC3
CL 0.5646223 -0.2405906 -0.7895048
CW 0.5475633 -0.6065478  0.5764323
BD 0.6175566  0.7577704  0.2107318

> summary(log.pcx)

Importance of components:
                          PC1     PC2     PC3
Standard deviation     0.4174 0.04387 0.01042
Proportion of Variance 0.9885 0.01092 0.00062
Cumulative Proportion  0.9885 0.99938 1.00000

> barplot(summary(log.pcx)$importance[2,],ylim=c(0,1))
> title(main="Proportion of variation",line=1)
```

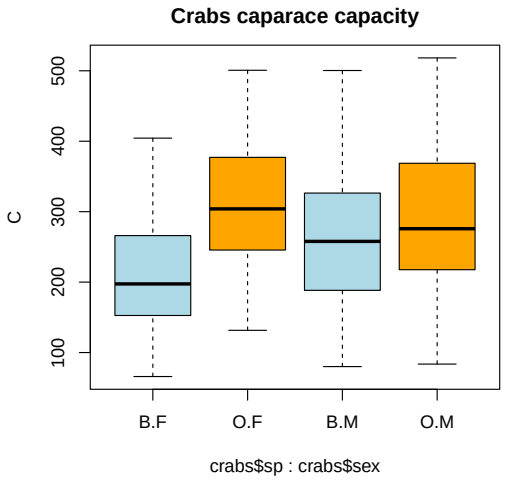


Como era de esperar las dos primeras componentes representan muy bien el conjunto de la variabilidad y la primera se puede interpretar como un factor tamaño. Entonces

$$\log(C) = 0.565\log(CL) + 0.548\log(CW) + 0.618\log(BD)$$

es decir

```
> logC <- as.matrix(log(Xc)) %*% log.pcx$rotation[,1]
> C <- exp(logC)
> # Representar las distribuciones por sexo y especie de la capacidad
> boxplot(C ~ crabs$sp + crabs$sex,
+                   col=c("lightblue","orange","lightblue","orange"))
> title(main="Crabs caparace capacity",line=1)
```



Parece que la especie Orange tiene una mayor capacidad del caparazón en los dos sexos.

Ejercicio 5

Florence Nightingale (1820-1910) fué una enfermera británica pionera en la aplicación de la estadística a la epidemiología. Entre otras acciones, ella recogió los datos de soldados muertos por diversas causas en la guerra de Crimea y convenció con sus argumentos estadísticos a las autoridades militares para modificar el sistema hospitalario hacia un modelo con más higiene.

Figura 3: Florence Nightingale (1820-1910).

La tabla de los datos en los 24 meses observados puede obtenerse en la página

`http://understandinguncertainty.org/node/214`

donde los 12 primeros son antes de aplicar sus nuevos métodos de cuidado en los hospitales militares. Estos mismos datos se hallan en el data.frame `Nightingale` *del paquete* `HistData`*.*
A partir de las frecuencias de muertes por tres causas: Zymotic diseases, Wounds & injuries y All other causes, Nightingale calculó unos índices de mortalidad anual relativa por cada 1000. Son precisamente esos índices los que vamos a utilizar como variables en este ejercicio.

a) *Observar e interpretar las representaciones descriptivas de la página*

`https://www.florence-nightingale.co.uk/coxcomb-diagram-1858/`

con el llamado diagrama de Coxcomb o rosa de Nightingale.

Parece que Florence Nightingale es la responsable del término "Estadística aplicada" que se introdujo en el currículum de la educación superior gracias a su influencia en Francis Galton y Karl Pearson.

Entre otras aportaciones, Nightingale utilizó una novedosa forma gráfica para la época basada en gráficos circulares para influir en la opinión pública y las autoridades británicas a fin de modificar la organización sanitaria y las condiciones de los hospitales británicos y de sus colonias en el mundo. Hoy se lo conoce como *histograma circular* o *diagrama de área polar*.

El diagrama de área polar es similar a un gráfico circular normal, a excepción de los sectores son ángulos iguales y se diferencian más bien en qué medida cada sector se extiende desde el centro del

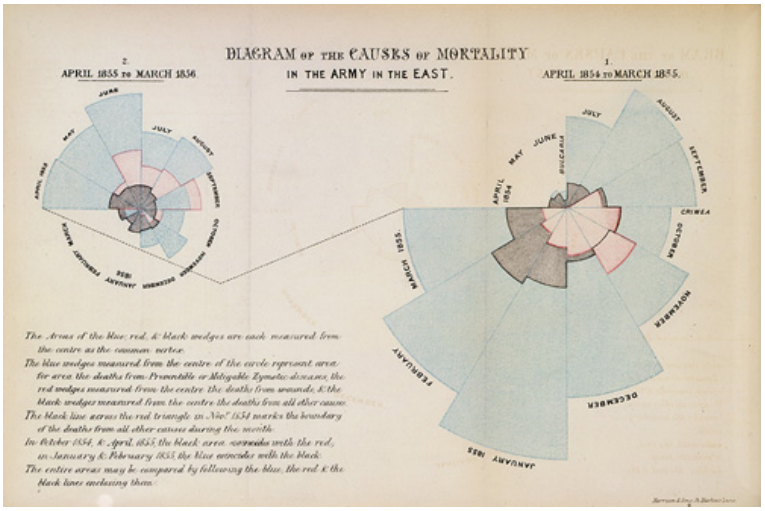


Figura 4: Nightingale's coxcombs.

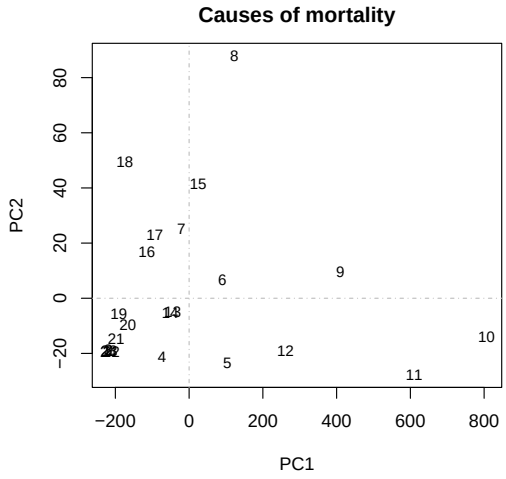
círculo. El diagrama de área polar se utiliza para representar los fenómenos cíclicos. En la figura 4 cada sector es un mes y el círculo completo representa un año.

La diferencia en el tamaño de los dos diagramas muestra la reducción de las muertes en el período tras la aplicación de las reformas sanitarias.

b) *Realizar un análisis de componentes principales sobre los índices de mortalidad, representar los 24 meses con números correlativos e interpretar el resultado.*

```
> library(HistData)
> data("Nightingale")
> str(Nightingale)

'data.frame': 24 obs. of  10 variables:
 $ Date         : Date, format: "1854-04-01" "1854-05-01" ...
 $ Month        : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<..: 4 5 6 7 8 9 10 11 12 1 ...
 $ Year         : int  1854 1854 1854 1854 1854 1854 1854 1854 1854 1855 ...
 $ Army         : int  8571 23333 28333 28722 30246 30290 30643 29736 32779 32393 ...
 $ Disease      : int  1 12 11 359 828 788 503 844 1725 2761 ...
 $ Wounds       : int  0 0 0 0 1 81 132 287 114 83 ...
 $ Other        : int  5 9 6 23 30 70 128 106 131 324 ...
 $ Disease.rate: num  1.4 6.2 4.7 150 328.5 ...
 $ Wounds.rate : num  0 0 0 0 0.4 ...
 $ Other.rate  : num  7 4.6 2.5 9.6 11.9 27.7 50.1 42.8 48 120 ...

> X <- Nightingale[,8:10]
> pcx <- prcomp(X); pcx

Standard deviations (1, .., p=3):
[1] 275.23286  28.27266  14.60615

Rotation (n x k) = (3 x 3):
                    PC1         PC2        PC3
Disease.rate 0.99235992 -0.01323725  0.1226644
```

```
Wounds.rate  0.02594605  0.99438465 -0.1025962
Other.rate   0.12061755 -0.10499500 -0.9871309

> # Representación de los datos en componentes principales
> plot(pcx$x[,1:2],pch="",xlab="PC1",ylab="PC2")
> abline(h=0,v=0,lty=4,col="gray")
> text(pcx$x[,1],pcx$x[,2],labels=1:24,cex=0.8)
> title(main="Causes of mortality",line=1)
```



Cada componente principal está asociada a una de las variables, pero la primera tiene todos sus coeficientes positivos de forma que se puede considerar como el tamaño o medida de las muertes.

La segunda componente se basa especialmente en la proporción de muertes por heridas o lesiones de forma que la parte superior del gráfico se hallan los meses con mayor ratio en estas causas.

Si observamos los números correlativos de los meses, los números iniciales (del 1 al 12) se sitúan mayoritariamente a la derecha, mientras que los números del segundo año (tras la reforma) están todos a la izquierda (excepto el 15). Eso muestra la reducción de muertes en el segundo período gracias a la reforma sanitaria.

Ejercicio 6

Con los moluscos gasterópodos conocidos como orejas de mar o abulones[3]*, muy apreciados por su carne, se estima la edad contando el número de anillos y es un trabajo muy laborioso. Por ello se busca la forma de predecir la edad de un abulón utilizando otras medidas más sencillas de obtener.*
Con el siguiente código se pueden obtener los datos de un conjunto de abulones pertenecientes a un estudio realizado por el Departamento de Industria y Pesca de Tasmania (Australia) en 1994:

```
> archivo <-
+ "http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data"
> abalone <- read.csv(archivo, header=F)
> names(abalone) <-
+        c("Sex",              # Sexo: M, F, and I (infant)
+          "Length",           # Longitud: mayor medida de la concha (mm)
+          "Diameter",         # Diámetro: perpendicular a la longitud (mm)
```

[3] Ver la página web `http://es.wikipedia.org/wiki/Haliotis`

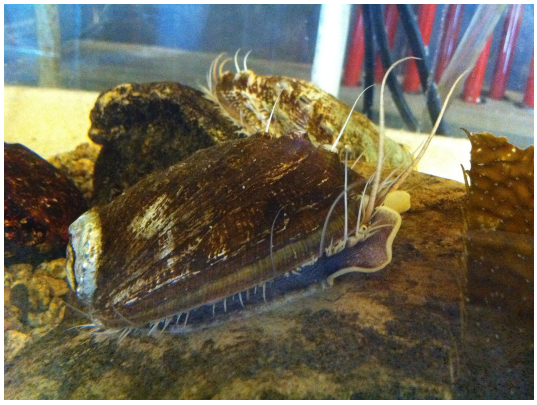


Figura 5: Living abalone showing epipodium and tentacles.

```
+           "Height",           # Altura: con carne dentro de la concha (mm)
+           "Whole weight",     # Peso total: todo el abulón (g)
+           "Shucked weight",   # Peso desconchado: peso de la carne (g)
+           "Viscera weight",   # Peso de las vísceras: peso de la tripa
+                               # (después de sangrar) (g)
+           "Shell weight",     # Peso de la concha: después de ser secado (g)
+           "Rings")            # Anillos: +1.5 es la edad en años
```

a) *Seleccionar un conjunto de 30 abulones hembra al azar*[4] *y realizar con ellos un análisis de componentes principales con todas las variables numéricas excepto la última (los anillos).*

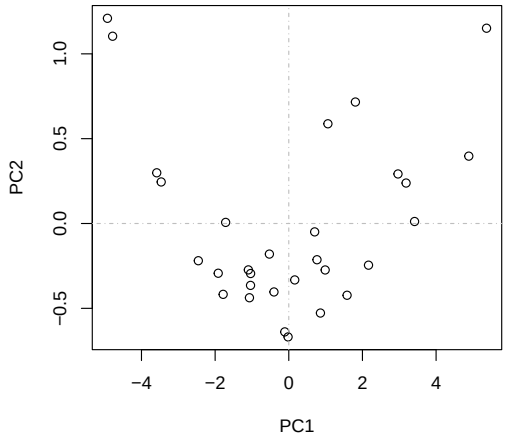
```
> set.seed(123)
> abalone.F <- abalone[abalone$Sex=="F", -1]
> aux <- sample(1:dim(abalone.F)[1],size = 30)
> mis.orejas <- abalone.F[aux, 1:7]
> pca <- prcomp(mis.orejas, scale.=TRUE, rank. = 4); pca

Standard deviations (1, .., p=7):
[1] 2.4658602 0.6062282 0.4887487 0.3930752 0.3426509 0.1502856 0.1365357

Rotation (n x k) = (7 x 4):
                      PC1          PC2        PC3        PC4
Length         -0.3881726 -0.336168534  0.1562581 -0.3867187
Diameter       -0.3855480 -0.337389793  0.1898031 -0.4555930
Height         -0.3391507  0.865835718  0.2525811 -0.2373304
Whole weight   -0.3994576  0.023045511 -0.0478777  0.2836769
Shucked weight -0.3747568  0.032938195 -0.6269156 -0.1890481
Viscera weight -0.3797193  0.002598431 -0.4396929  0.4245772
Shell weight   -0.3761060 -0.147866770  0.5358317  0.5385911

> summary(pca)

Importance of first k=4 (out of 7) components:
                          PC1    PC2     PC3     PC4
```

[4]Se puede utilizar la función `sample()` de **R**.

```
Standard deviation     2.4659 0.6062 0.48875 0.39308
Proportion of Variance 0.8686 0.0525 0.03413 0.02207
Cumulative Proportion  0.8686 0.9211 0.95526 0.97734
```

Representar estos abulones con puntos en un diagrama de dispersión formado por los dos primeros ejes principales.

```
> plot(pca$x[,1:2],xlab="PC1",ylab="PC2")
> abline(h=0,v=0,lty=4,col="gray")
```



b) *Para los primeros 151 abulones, obtener la matriz de correlaciones de las variables numéricas sin la última (los anillos).*

£Qué problema tendremos si pretendemos predecir la edad del abulón con estas variables?

```
> cor(abalone[1:151,2:8])

                  Length  Diameter    Height Whole weight Shucked weight
Length         1.0000000 0.9889154 0.9145329    0.9233945      0.9218379
Diameter       0.9889154 1.0000000 0.9255887    0.9285294      0.9227633
Height         0.9145329 0.9255887 1.0000000    0.8996011      0.8814904
Whole weight   0.9233945 0.9285294 0.8996011    1.0000000      0.9790210
Shucked weight 0.9218379 0.9227633 0.8814904    0.9790210      1.0000000
Viscera weight 0.9016638 0.9058299 0.8979068    0.9350467      0.9455367
Shell weight   0.8821797 0.8896391 0.8663760    0.9688221      0.9148738
               Viscera weight Shell weight
Length              0.9016638    0.8821797
Diameter            0.9058299    0.8896391
Height              0.8979068    0.8663760
Whole weight        0.9350467    0.9688221
Shucked weight      0.9455367    0.9148738
Viscera weight      1.0000000    0.8500491
Shell weight        0.8500491    1.0000000
```

Podría pensarse en la regresión de la variable dependiente anillos, respecto a las variables independientes, o predictoras, recogidas en este caso, que representan medidas relacionadas todas ellas con

la edad del abulón. El problema es que están muy correlacionadas entre sí. El aumento o disminución en el valor de una, implica el aumento o disminución en cualquiera de las otras.

En este caso se presenta un problema conocido como de multicolinealidad, por el que los coeficientes de regresión quedan indeterminados en alto grado debido a la dependencia entre las variables explicativas. Esto provoca un aumento en la varianza de los coeficientes de regresión, lo que a su vez aumenta los intervalos de confianza de las predicciones, haciéndolas muy imprecisas. Por otra parte precluye interpretaciones respecto a los coeficientes de las variables en términos de fijar las demás y aumentar la variable en una unidad, ya que el aumento en una variable implica variaciones implícitas en las otras variables correlacionadas.

c) *Para paliar el problema anterior, realizar una regresión sobre la variable* `Rings` *con las dos primeras componentes principales obtenidas con la matriz de covarianzas.*

Estudiar la bondad de este modelo, frente al modelo de regresión con las variables originales.

Hay que hacer algunas pruebas con el modelo de regresión con las variables originales, pero en el análisis de los residuos se observa que el dato 2052 es un dato influyente de forma exagerada de manera que lo eliminamos del estudio.

Otro problema, como ya se ha dicho, es la multicolinealidad que se puede calcular con los VIF. Todo VIF superior a 10 (incluso a 5) se considera problemático.

```
> library(car)
> lmod <- lm(Rings ~ ., data=abalone[-2052,2:9])
> summary(lmod)


Call:
lm(formula = Rings ~ ., data = abalone[-2052, 2:9])

Residuals:
    Min      1Q  Median      3Q     Max
-8.0883 -1.3643 -0.3699  0.9055 13.9260

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       2.7049     0.2703  10.005  < 2e-16 ***
Length           -2.0547     1.8151  -1.132    0.258
Diameter         11.6333     2.2366   5.201 2.07e-07 ***
Height           22.9020     2.1796  10.507  < 2e-16 ***
`Whole weight`    9.1266     0.7284  12.529  < 2e-16 ***
`Shucked weight` -19.8946     0.8196 -24.274  < 2e-16 ***
`Viscera weight` -10.3755     1.2984  -7.991 1.72e-15 ***
`Shell weight`    7.6632     1.1371   6.740 1.81e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.204 on 4168 degrees of freedom
Multiple R-squared:  0.5334,Adjusted R-squared:  0.5326
F-statistic: 680.5 on 7 and 4168 DF,  p-value: < 2.2e-16

> vif(lmod)

          Length         Diameter           Height   `Whole weight`
       40.824692        42.333132         6.182576       109.645321
`Shucked weight` `Viscera weight`   `Shell weight`
       28.436936        17.404570        21.524931
```

El modelo con las dos primeras componentes es

```
> pca <- prcomp(abalone[-2052,2:8], rank. = 4)
> lmodpcr <- lm(abalone$Rings[-2052] ~ pca$x[,1] + pca$x[,2])
> summary(lmodpcr)


Call:
lm(formula = abalone$Rings[-2052] ~ pca$x[, 1] + pca$x[, 2])

Residuals:
    Min      1Q  Median      3Q     Max
-6.9575 -1.4051 -0.4171  0.8906 15.2767

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.93415    0.03508  283.17   <2e-16 ***
pca$x[, 1]   2.96925    0.06033   49.22   <2e-16 ***
pca$x[, 2]  23.97001    0.55724   43.02   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.267 on 4173 degrees of freedom
Multiple R-squared:  0.5059,Adjusted R-squared:  0.5057
F-statistic:  2136 on 2 and 4173 DF,  p-value: < 2.2e-16
```

Como sabemos, las componentes son ortogonales y el problema de la multicolinealidad no existe. El ajuste, comparado con el modelo anterior, es razonable y podría ser mejor con más componentes.

Este modelo presenta otras dificultades, que también habría que tratar, como la heterocedasticidad.

También sería interesante estudiar la importancia de las variables en la regresión y su interpretación en el modelo, para utilizar aquellas que realmente se necesitan.

La regresión con componentes principales o PCR (*Principal Components Regression*) se puede hacer con la función `pcr()` del paquete `pls`.

Ejercicio 7

Para la base de datos `usair` *del libro de Everitt(2005)*[5]

a) *Cargar los datos con la instrucción:*

```
> usair <- source("chap3usair.dat")$value
```

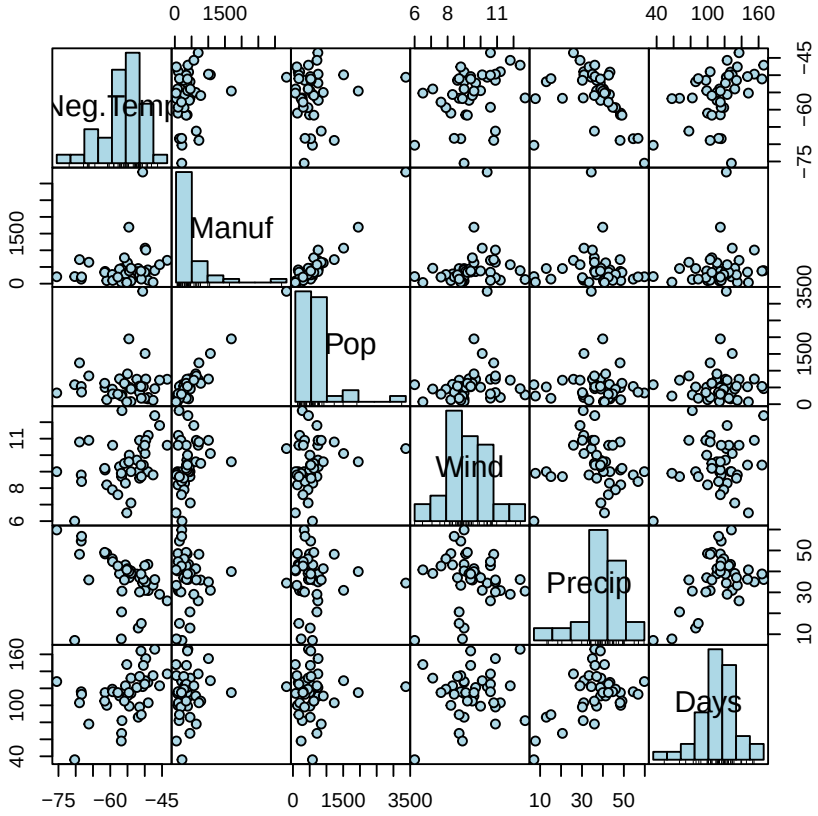
y estudiar la base de datos con las funciones `str()` *y* `summary()`.

Realizar un gráfico descriptivo de los datos.

```
> str(usair)

'data.frame': 41 obs. of  7 variables:
 $ SO2     : num  10 13 12 17 56 36 29 14 10 24 ...
 $ Neg.Temp: num  -70.3 -61 -56.7 -51.9 -49.1 -54 -57.3 -68.4 -75.5 -61.5 ...
```

[5] Ver la página web `http://biostatistics.iop.kcl.ac.uk/publications/everitt/`.

```
 $ Manuf   : num  213 91 453 454 412 80 434 136 207 368 ...
 $ Pop     : num  582 132 716 515 158 80 757 529 335 497 ...
 $ Wind    : num  6 8.2 8.7 9 9 9 9.3 8.8 9 9.1 ...
 $ Precip  : num  7.05 48.52 20.66 12.95 43.37 ...
 $ Days    : num  36 100 67 86 127 114 111 116 128 115 ...

> summary(usair)

      SO2            Neg.Temp          Manuf             Pop
 Min.   :  8.00   Min.   :-75.50   Min.   :  35.0   Min.   :  71.0
 1st Qu.: 13.00   1st Qu.:-59.30   1st Qu.: 181.0   1st Qu.: 299.0
 Median : 26.00   Median :-54.60   Median : 347.0   Median : 515.0
 Mean   : 30.05   Mean   :-55.76   Mean   : 463.1   Mean   : 608.6
 3rd Qu.: 35.00   3rd Qu.:-50.60   3rd Qu.: 462.0   3rd Qu.: 717.0
 Max.   :110.00   Max.   :-43.50   Max.   :3344.0   Max.   :3369.0
      Wind            Precip           Days
 Min.   : 6.000   Min.   : 7.05   Min.   : 36.0
 1st Qu.: 8.700   1st Qu.:30.96   1st Qu.:103.0
 Median : 9.300   Median :38.74   Median :115.0
 Mean   : 9.444   Mean   :36.77   Mean   :113.9
 3rd Qu.:10.600   3rd Qu.:43.11   3rd Qu.:128.0
 Max.   :12.700   Max.   :59.80   Max.   :166.0

> # Matriz de correlaciones entre variables
> round(cor(usair),4)

            SO2 Neg.Temp   Manuf     Pop    Wind  Precip   Days
SO2      1.0000   0.4336  0.6448  0.4938  0.0947  0.0543 0.3696
Neg.Temp 0.4336   1.0000  0.1900  0.0627  0.3497 -0.3863 0.4302
Manuf    0.6448   0.1900  1.0000  0.9553  0.2379 -0.0324 0.1318
Pop      0.4938   0.0627  0.9553  1.0000  0.2126 -0.0261 0.0421
Wind     0.0947   0.3497  0.2379  0.2126  1.0000 -0.0130 0.1641
Precip   0.0543  -0.3863 -0.0324 -0.0261 -0.0130  1.0000 0.4961
Days     0.3696   0.4302  0.1318  0.0421  0.1641  0.4961 1.0000

> # Función de panel diagonal: histograma y densidad
> dgp.fn <- function(x,...){
+   par(new=TRUE)
+   hist(x,col="lightblue",probability=TRUE,axes=FALSE,main="")
+   rug(x)
+ }
> # Función de panel no diagonal: diagrama dispersión y recta de regresión
> pn.fn <- function(x,y,...) points(x,y,pch=21,bg="lightblue")
> # Matriz de diagramas de dispersión
> pairs(usair[,2:7],panel=pn.fn,diag.panel=dgp.fn,
+       label.pos=0.5,cex.labels=1.5, gap=0)
```



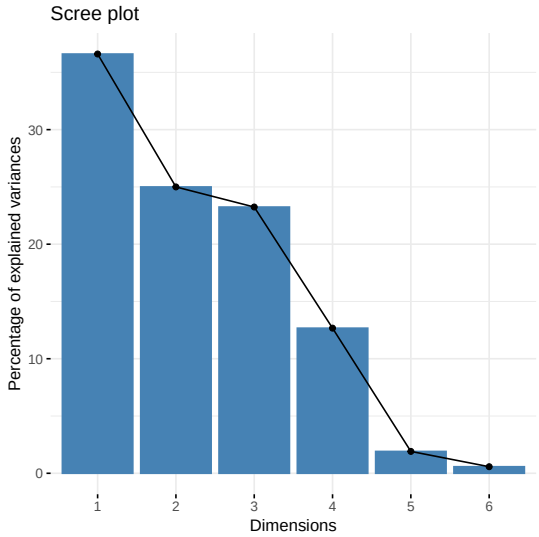
b) *Representar las ciudades en el plano de las dos primeras componentes principales calculadas con la matriz de correlaciones de todas las variables excepto el* `SO2`.

Estudiar las posibles ciudades atípicas (outliers).

```
> pca <- prcomp(usair[,-1],scale=TRUE)
> round(pca$rot, 4)

             PC1     PC2     PC3     PC4     PC5     PC6
Neg.Temp  0.3296 -0.1276  0.6717  0.3065 -0.5581  0.1362
Manuf     0.6115  0.1681 -0.2729  0.1368 -0.1020 -0.7030
Pop       0.5778  0.2225 -0.3504  0.0725  0.0781  0.6946
Wind      0.3538 -0.1308  0.2973 -0.8694  0.1133 -0.0245
Precip   -0.0408 -0.6229 -0.5046 -0.1711 -0.5682  0.0606
Days      0.2379 -0.7078  0.0931  0.3113  0.5800 -0.0220

> summary(pca)

Importance of components:
                          PC1   PC2    PC3    PC4    PC5     PC6
Standard deviation     1.482 1.225 1.1810 0.8719 0.3385 0.18560
Proportion of Variance 0.366 0.250 0.2324 0.1267 0.0191 0.00574
Cumulative Proportion  0.366 0.616 0.8485 0.9752 0.9943 1.00000
```

Con el paquete `factoextra` se obtienen algunos gráficos mejorados.

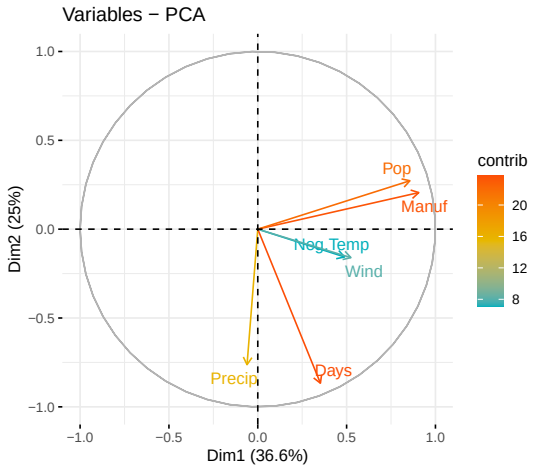
```
> library(factoextra)
> fviz_eig(pca)
```



En este análisis se necesitan tres componentes principales para superar el 80 % de variabilidad explicada y cuatro componentes para superar el 90 %.

Para interpretar las componentes podemos utilizar el siguiente gráfico de las variables.

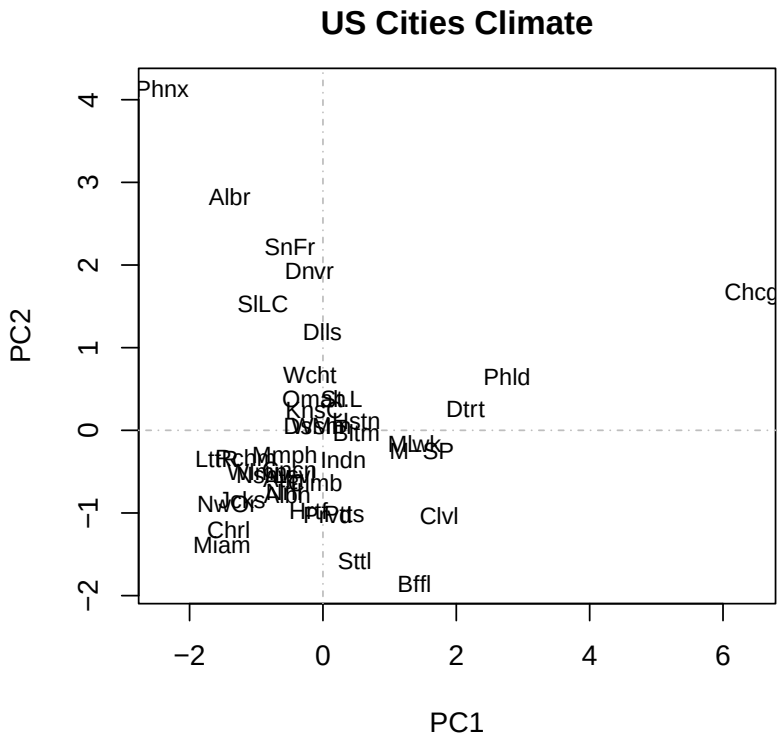
```
> fviz_pca_var(pca,
+              col.var = "contrib", # Color by contributions to the PC
+              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
+              repel = TRUE     # Avoid text overlapping
+              )
```



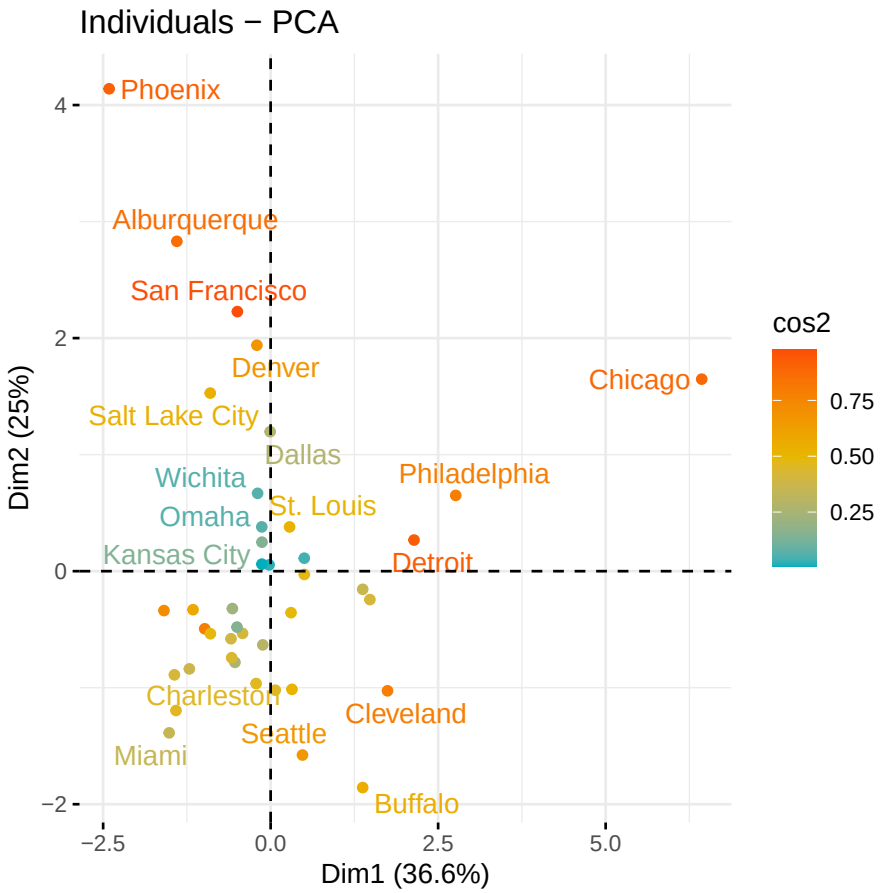
En la primera componente, la variable más importante es `Manuf` y, aunque hay un coeficiente negativo, la PC1 se puede interpretar como el "tamaño" de la ciudad. La segunda componente contrapone las variables meteorológicas a las socioeconómicas.

El gráfico de las ciudades es

```
> plot(pca$x[,1:2],pch="",xlab="PC1",ylab="PC2")
> abline(h=0,v=0,lty=4,col="gray")
> text(pca$x[,1],pca$x[,2],labels=abbreviate(row.names(usair)),cex=0.8)
> title(main="US Cities Climate",line=1)
```




```
> fviz_pca_ind(pca,
+              col.ind = "cos2", # Color by the quality of representation
+              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
+              repel = TRUE     # Avoid text overlapping
+              )
```



En la representación de los datos sobre el plano de las dos primeras componentes principales aparece la ciudad de Chicago como atípica con un valor muy grande para la primera componente. Eso significa que es una ciudad con valores grandes de las variables socioeconómicas y meteorológicas.

Además, también se observa Phoenix como otra posible ciudad atípica. En este caso por tener una segunda componente muy grande, lo que quiere decir que es una ciudad con valores grandes socioeconómicos y buen tiempo.

c) *Utilizar una matriz de correlaciones robusta*[6] *y calcular de nuevo las componentes principales. Comparar los resultados con los del apartado anterior.*

Utilizaremos una matriz de correlaciones robusta para minimizar la influencia de observaciones atípicas sobre el resto, lo que permite conservarlas a todas.

```
> # Correlación de Spearman
> X <- usair[,-1]
> cor.sp <- cor(X, method="spearman"); round(cor.sp,4)

          Neg.Temp   Manuf     Pop    Wind  Precip   Days
Neg.Temp   1.0000  0.2256 -0.0314  0.3983 -0.4986 0.5057
Manuf      0.2256  1.0000  0.8230  0.3942 -0.1516 0.1453
Pop       -0.0314  0.8230  1.0000  0.3374 -0.1303 0.0101
Wind       0.3983  0.3942  0.3374  1.0000 -0.2566 0.0345
```

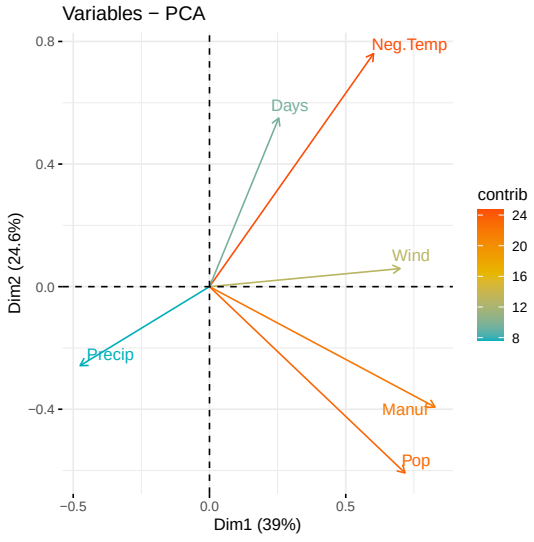
[6] Como coeficiente de correlación robusto se puede calcular la rho de Spearman o la tau de Kendall. (∗) También se puede utilizar una estimación robusta de la matriz de correlaciones por el método del elipsoide de mínimo volumen.

```
Precip    -0.4986 -0.1516 -0.1303 -0.2566  1.0000 0.2423
Days       0.5057  0.1453  0.0101  0.0345  0.2423 1.0000
```

Para el análisis de las componentes principales utilizaremos la función `princomp()` del paquete `MASS`.

```
> pca.sp <- princomp(covmat=cor.sp, cor=TRUE)
> summary(pca.sp)

Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4    Comp.5
Standard deviation     1.5303805 1.2141863 1.1189397 0.8076990 0.4410197
Proportion of Variance 0.3903441 0.2457081 0.2086710 0.1087296 0.0324164
Cumulative Proportion  0.3903441 0.6360521 0.8447231 0.9534527 0.9858691
                           Comp.6
Standard deviation     0.29117895
Proportion of Variance 0.01413086
Cumulative Proportion  1.00000000

> round(pca.sp$loadings[,1:6],4)

          Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6
Neg.Temp  0.3935  0.6262  0.0346  0.1039  0.3486  0.5653
Manuf     0.5409 -0.3236 -0.1999  0.2051  0.5939 -0.4098
Pop       0.4688 -0.5005 -0.1435  0.1977 -0.4756  0.4938
Wind      0.4571  0.0486  0.1435 -0.8541 -0.1409 -0.1368
Precip   -0.3102 -0.2122 -0.6743 -0.4040  0.3300  0.3631
Days      0.1661  0.4530 -0.6804  0.1236 -0.4132 -0.3438
```

```
> fviz_pca_var(pca.sp,
+              col.var = "contrib", # Color by contributions to the PC
+              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
+              repel = TRUE     # Avoid text overlapping
+              )
```



Ahora, en la primera componente destacan las mismas variables pero con signo contrario. La segunda componente es bastante distinta al análisis anterior.

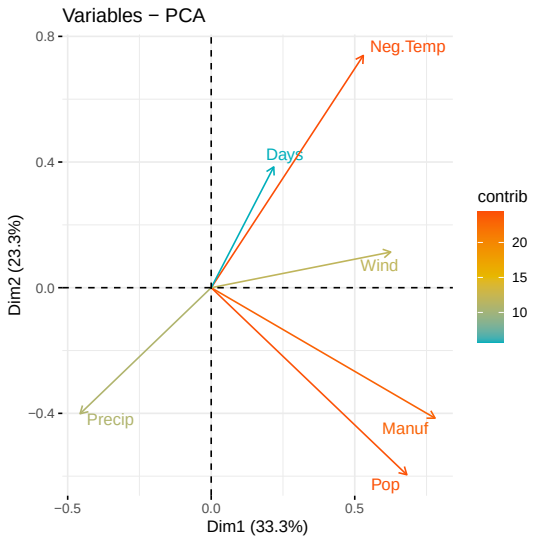
Repetimos el análisis con la matriz de correlaciones tau de Kendall.

```
> # Correlación de Kendall
> X <- usair[,-1]
> cor.knd <- cor(X, method="kendall"); round(cor.knd,4)

         Neg.Temp   Manuf     Pop    Wind  Precip   Days
Neg.Temp   1.0000  0.1416 -0.0342  0.2910 -0.4029 0.3534
Manuf      0.1416  1.0000  0.6585  0.2808 -0.1268 0.1054
Pop       -0.0342  0.6585  1.0000  0.2340 -0.0829 0.0098
Wind       0.2910  0.2808  0.2340  1.0000 -0.1946 0.0198
Precip    -0.4029 -0.1268 -0.0829 -0.1946  1.0000 0.1544
Days       0.3534  0.1054  0.0098  0.0198  0.1544 1.0000
```

Para el análisis de las componentes principales utilizaremos también la función `princomp()`.

```
> pca.knd <- princomp(covmat=cor.knd, cor=TRUE)
> summary(pca.knd)

Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
Standard deviation     1.4144192 1.1811401 1.0792049 0.8701285 0.61146689
Proportion of Variance 0.3334303 0.2325153 0.1941139 0.1261873 0.06231529
Cumulative Proportion  0.3334303 0.5659456 0.7600595 0.8862467 0.94856203
                           Comp.6
Standard deviation     0.55554280
Proportion of Variance 0.05143797
Cumulative Proportion  1.00000000

> round(pca.knd$loadings[,1:6],4)

          Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6
Neg.Temp  0.3746  0.6258  0.0706  0.1024  0.5452  0.3941
Manuf     0.5514 -0.3517  0.1357  0.1906  0.4025 -0.5963
Pop       0.4818 -0.5041  0.0702  0.1769 -0.2581  0.6411
Wind      0.4422  0.0961 -0.1463 -0.8579 -0.1791 -0.0761
Precip   -0.3229 -0.3397  0.5887 -0.4199  0.4608  0.2126
Days      0.1540  0.3255  0.7770  0.0981 -0.4794 -0.1649
```

```
> fviz_pca_var(pca.knd,
+              col.var = "contrib", # Color by contributions to the PC
+              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
+              repel = TRUE     # Avoid text overlapping
+              )
```



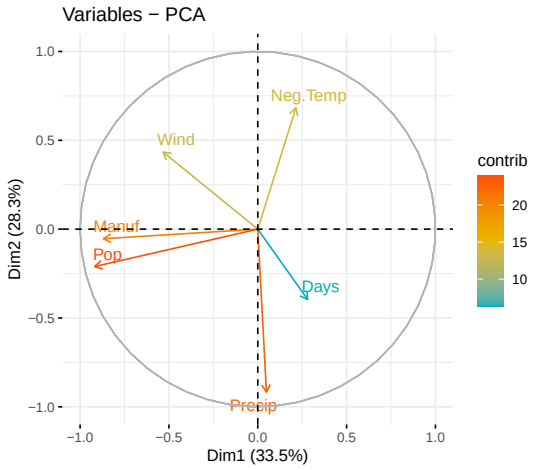
Aunque la representación es un poco peor, el gráfico y la interpretación es la misma que para la correlación rho de Spearman.

Finalmente vamos a utilizar la correlación robusta por el método del *minimum volume ellipsoid*.

```
> # Correlación robusta por el método del minimum volume ellipsoid
> usair.mve <- cov.mve(X,cor=TRUE)
> round(usair.mve$cor,4)

          Neg.Temp   Manuf     Pop    Wind  Precip    Days
Neg.Temp   1.0000  0.0178 -0.2665  0.1559 -0.5347  0.3429
Manuf      0.0178  1.0000  0.8074  0.2686 -0.0125  0.0200
Pop       -0.2665  0.8074  1.0000  0.2800  0.0802 -0.0767
Wind       0.1559  0.2686  0.2800  1.0000 -0.1862 -0.2518
Precip    -0.5347 -0.0125  0.0802 -0.1862  1.0000  0.4543
Days       0.3429  0.0200 -0.0767 -0.2518  0.4543  1.0000

> pca.mve <- princomp(covmat=usair.mve, cor=TRUE)
> summary(pca.mve)

Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
Standard deviation     1.4169844 1.3036739 1.1504340 0.8537485 0.41785631
Proportion of Variance 0.3346408 0.2832610 0.2205831 0.1214811 0.02910065
Cumulative Proportion  0.3346408 0.6179018 0.8384848 0.9599659 0.98906653
                           Comp.6
Standard deviation     0.25612653
Proportion of Variance 0.01093347
Cumulative Proportion  1.00000000

> round(pca.mve$loadings[,1:6],4)

          Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6
Neg.Temp  0.1515  0.5237  0.5912  0.0194  0.1278  0.5801
Manuf    -0.6130 -0.0414  0.2977  0.2660  0.6315 -0.2539
Pop      -0.6479 -0.1614  0.0921  0.1897 -0.6214  0.3516
```

```
Wind     -0.3758  0.3327 -0.0090 -0.8498 -0.0432 -0.1551
Precip    0.0343 -0.7037  0.0780 -0.3971  0.3150  0.4906
Days      0.1980 -0.3035  0.7397 -0.1146 -0.3126 -0.4589
```

```
> fviz_pca_var(pca.mve,
+              col.var = "contrib", # Color by contributions to the PC
+              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
+              repel = TRUE     # Avoid text overlapping
+              )
```



En este último análisis y en la primera componente, la variable `Precip` sí es importante y en sentido contrario a las otras. En la segunda componente pierde importancia la variable `Days`. Así pues, el resultado será distinto a los dos análisis robustos anteriores.

d) (∗) *Investigar la utilización de la regresión múltiple con las variables del clima y poblacionales para predecir la polución en dióxido de sulfuro, teniendo en cuenta el problema de alta correlación entre algunas variables predictoras.*

En primer lugar haremos una selección de variables por el método stepwise.

```
> g <- lm(SO2~.,data=usair)
> formula(step(g))

Start:  AIC=226.37
SO2 ~ Neg.Temp + Manuf + Pop + Wind + Precip + Days

           Df Sum of Sq     RSS    AIC
- Days      1      22.1  7305.4 224.50
<none>                   7283.3 226.37
- Precip    1     427.3  7710.6 226.71
- Wind      1     658.1  7941.4 227.92
- Neg.Temp  1     892.5  8175.8 229.11
- Pop       1    1443.1  8726.3 231.78
- Manuf     1    3640.1 10923.4 240.99
```

```
Step:  AIC=224.49
SO2 ~ Neg.Temp + Manuf + Pop + Wind + Precip

           Df Sum of Sq     RSS    AIC
<none>                  7305.4 224.50
- Wind      1     636.1  7941.5 225.92
- Precip    1     785.4  8090.8 226.68
- Pop       1    1447.5  8752.9 229.91
- Neg.Temp  1    1517.4  8822.8 230.23
- Manuf     1    3636.8 10942.1 239.06
SO2 ~ Neg.Temp + Manuf + Pop + Wind + Precip
```

Sólo se elimina la variable `Days`, pero se conservan variables que están correlacionadas.

```
> g2 <- update(g, .~.-Days)
> vif(g2)

 Neg.Temp     Manuf       Pop      Wind    Precip
 1.731366 14.703099 14.338797  1.219354  1.241777

> g3 <- update(g2, .~.-Pop)
> vif(g3)

Neg.Temp    Manuf     Wind   Precip
1.384555 1.075237 1.202426 1.199748
```

Otra posibilidad es hacer una Principal Components Regression con tres o cuatro componentes.

```
> library(pls)
> g4 <- pcr(SO2~.,data=usair, ncomp=4)
> summary(g4)

Data:  X dimension: 41 6
Y dimension: 41 1
Fit method: svdpc
Number of components considered: 4
TRAINING: % variance explained
     1 comps  2 comps  3 comps  4 comps
X      97.63    99.88    99.98   100.00
SO2    32.91    58.91    61.74    62.04

> coef(g4)

, , 4 comps

                SO2
Neg.Temp  0.07144595
Manuf     0.07229766
Pop      -0.04761270
Wind      0.00355861
Precip   -0.05805350
Days      0.17592362
```

También podemos utilizar las componentes principales halladas con una matriz de correlaciones robusta como la que hemos obtenido con la rho de Spearman.

```
> x <- scale(as.matrix(usair[,2:7]),scale=TRUE) %*% pca.sp$loadings
> g5 <- lm(usair$SO2 ~ x[,1] + x[,2] + x[,3] + x[,4])
> summary(g5)


Call:
lm(formula = usair$SO2 ~ x[, 1] + x[, 2] + x[, 3] + x[, 4])

Residuals:
    Min      1Q  Median      3Q     Max
-30.835  -9.685  -2.412   9.589  66.028

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   30.049      2.702  11.122 3.36e-13 ***
x[, 1]         8.628      1.906   4.527 6.31e-05 ***
x[, 2]         1.386      2.358   0.588  0.56029
x[, 3]        -6.946      2.290  -3.033  0.00447 **
x[, 4]         7.367      3.064   2.404  0.02147 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.3 on 36 degrees of freedom
Multiple R-squared:  0.5112,Adjusted R-squared:  0.4568
F-statistic: 9.411 on 4 and 36 DF,  p-value: 2.593e-05
```

En estos dos últimos casos, aunque ganamos en precisión, perdemos en capacidad de interpretación del modelo.