

Análisis de componentes principales.



Reduciendo la dimensión de la información económico-financiera sobre las empresas eólicas.

Vamos a considerar una serie de variables (escala **métrica**) que caracterizan a la población de empresas de producción eléctrica mediante tecnología eólica, de la que hemos seleccionado una muestra de 60 empresas. Nuestro objetivo es seleccionar una serie de combinaciones lineales de estas variables (cada combinación es una “componente principal”) que recojan la mayor parte de la suma de varianzas de las variables originales (“**comunalidad**”), y que sean menos numerosas que las variables originales, de manera que puedan usarse como variables que resumen o sintetizan la información que las variables originales, con una **pérdida mínima de información**, y además estando **incorrelacionadas** entre sí.

Preparando los datos. Buscando *missing values* y *outliers*.

Abriremos **R-Studio** y crearemos nuestro **proyecto** siguiendo la instrucción **File → New Project**. Luego nos preguntará si crea el proyecto en una nueva carpeta o en una ya existente. Vamos a crearlo, por ejemplo, en el disco extraíble D, carpeta R, subcarpeta “componentes”, que ya está creada. Nos saldrá una ventana para buscar la carpeta y, cuando la encontremos, pulsamos **Open** y **Create Project**.

Vamos a ir a la carpeta del proyecto y vamos a guardar en ella los dos archivos de esta práctica: un archivo de **Microsoft® Excel®** llamado “eolica_60.xlsx” y un *script* denominado “componentes_eolica.R”. Si abrimos el archivo de **Microsoft® Excel®**, comprobaremos que se compone de tres hojas. La primera muestra el criterio de búsqueda de casos en la base de datos **Sabi®**; la segunda recoge la descripción de las variables consideradas, y la tercera (hoja “Datos”) guarda los datos que debemos

importar desde **R-Studio**. Estos datos se corresponden con diferentes variables económico-financieras de 60 empresas productoras de electricidad mediante generación eólica.

Luego vamos a cerrar el archivo de **Microsoft® Excel®** y volveremos a **R-Studio**. Vamos a abrir nuestro script “componentes_eolica.R” con **File → Open File...** Este script contiene el programa que vamos a ir ejecutando en la práctica.

La primera línea / instrucción en el script es:

```
rm(list = ls())
```

La instrucción tiene como objeto limpiar el **Environment** de objetos de anteriores sesiones de trabajo. Para importar los datos, ejecutaremos el código:

```
# DATOS  
  
library(readxl)  
eolicos <- read_excel("eolica_60.xlsx", sheet = "Datos")
```

Podemos observar cómo, en el **Environment**, ya aparece un objeto. Este objeto es una estructura de datos tipo *data frame*, se llama “eolicos” y contiene 12 columnas, una por cada una de las variables almacenadas en el archivo de **Microsoft® Excel®**. De estas variables, tres son de tipo cualitativo (atributos o factores), formadas por cadenas de caracteres: el nombre de la empresa (NOMBRE), el nombre de la sociedad matriz (grupo empresarial) a la que pertenece (MATRIZ), y el tamaño de dicho grupo de empresas (DIMENSION).

R ha considerado la primera columna (NOMBRE) como una variable de tipo cualitativo. En realidad, no es una variable, sino el nombre de los casos (empresas). Para evitar que **R** tome los nombres de los individuos como una variable, podemos redefinir nuestro *data frame* diciéndole que tome esa primera columna como los nombres de los individuos o casos (filas):

```
eolicos <- data.frame(eolicos, row.names = 1)
```

En la línea anterior hemos asignado al *data frame* “eolicos” los propios datos de “eolicos”; pero indicando que la primera columna de datos no es una variable; sino el nombre de los individuos, casos o filas.

Vemos que ya no aparece NOMBRE como variable, y en el *Environment* ya aparece el *data frame* “eolicos” con 100 observaciones (casos), pero con 11 variables (una menos).

Por otro lado, el análisis de componentes principales solo se puede efectuar entre variables en escala métrica, por lo que, para seleccionar solo aquellas con tal naturaleza (todas, salvo MATRIZ y DIMENSION, que son las dos últimas columnas del *data frame* “eolicos”), crearemos un nuevo *data frame* llamado “originales”:

```
# Seleccionando variables metricas para el analisis

library(dplyr)
originales <- eolicos %>% select(eolicos, -MATRIZ, -DIMENSION)
summary (originales)
```

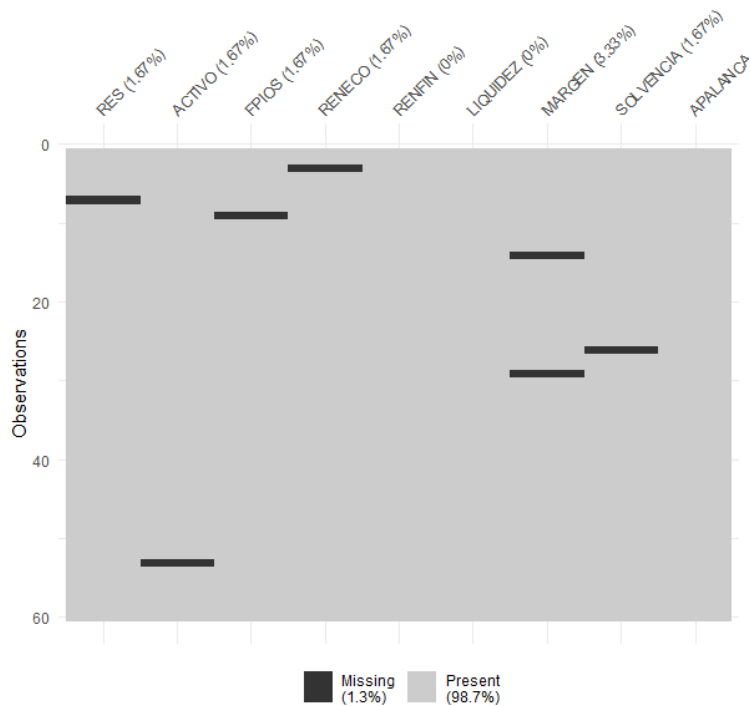
El siguiente paso será localizar los posibles *missing values*, ya que para obtener componentes principales es necesario que todos los casos posean dato para todas las variables del análisis.

Para tener una idea general, se puede utilizar la función *vis_miss()* del paquete *visdat*, que nos localizará gráficamente los *missing values* de las diferentes variables, y calculará el porcentaje de casos que supone, con respecto al total de observaciones:

```
# Identificando missing values.

library(visdat)
vis_miss(originales)
```

El resultado del código anterior es el siguiente gráfico:



Del gráfico anterior se desprende que existen 7 *missing values* repartidos en 6 de las 9 variables del estudio. Para localizarlos, podemos filtrar nuestro *data frame* con las herramientas de **dplyr**:

```
originales %>% filter(is.na(RES) | is.na(ACTIVO) | is.na(FPIOS) |
is.na(RENECO) | is.na(MARGEN) | is.na(SOLVENCIA)) %>%
select(RES, ACTIVO, FPIOS, RENECO, MARGEN, SOLVENCIA)
```

Los casos detectados con algún *missing value* son:

	RES	ACTIVO	FPIOS	RENECO	MARGEN	SOLVENCIA
Viesgo Renovables SL.	4609.00000	269730.00	177707.0000	NA	11.818	65.883
Biovent Energia SA	NA	183899.00	70033.0000	4.551	22.792	38.082
Eolica La Janda SL	9880.09100	153429.44	NA	8.586	38.256	16.428
Parc Eolic Sant Antoni SL	668.00000	69654.00	9727.0000	1.361	NA	13.964
WPD Parque Eolico El Poleo SL.	-30.63754	43997.60	520.6033	-0.092	-11.121	NA
Eolica La Brujula SA	2306.06200	42146.98	21694.7910	7.295	NA	51.474
Energias Naturales La Calzada SL	754.00000	NA	-1983.0000	3.446	15.561	-6.834

Ante la existencia de *missing values*, se puede actuar de varios modos. Por ejemplo, **se puede intentar obtener por otro canal de información el conjunto de valores** que no están disponibles, **o recurrir a alguna estimación**. En caso de que esto sea difícil, se puede optar, simplemente, por **eliminar** estos casos, en especial cuando representan un porcentaje muy reducido respecto al total de casos. En nuestro ejemplo, supondremos que hemos optado por esta última vía, y eliminaremos estos casos con el código:

```

originales <- originales %>%
  filter(! is.na(RES) & ! is.na(ACTIVO) & ! is.na(FPIOS) &
! is.na(RENECO) & ! is.na(MARGEN) & ! is.na(SOLVENCIA))

```

Podemos verificar en el *Environment* que el *data frame* “originales” ha pasado a tener 53 casos.

Por otro lado, la técnica de componentes principales **es muy sensible a la existencia de outliers**. En consecuencia, deberán ser identificados y, en su caso, eliminados. Para realizar este proceso, y dado que en nuestro análisis contamos con 9 variables, primero “resumiremos” el valor que toman dichas variables para cada caso, mediante el cálculo de la *distancia de Mahalanobis*. De hecho, las distancias de los diferentes casos se almacenarán en un vector, al que llamaremos “MAHALANOBIS”. Una vez calculado, lo añadiremos al *data frame* “originales_maha” mediante la función de pegado de columnas, `cbind()`:

```

# Identificando y eliminando outliers.

MAHALANOBIS <- mahalanobis(originales[,
  center = colMeans(originales[]),
  cov=cov(originales[]))
originales_maha <- cbind(originales, MAHALANOBIS)

```

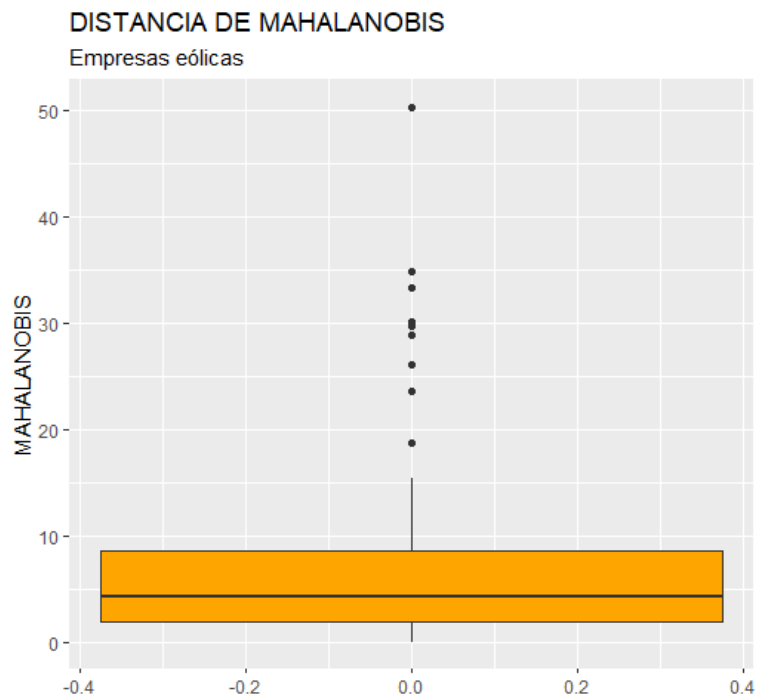
A continuación, construiremos un diagrama de caja de la variable MAHALANOBIS, como cualquier otra variable, a partir de la función `ggplot()` del paquete `ggplot2`:

```

library (ggplot2)
ggplot(data = originales_maha, map = (aes(y = MAHALANOBIS))) +
  geom_boxplot(fill = "orange") +
  ggtitle("DISTANCIA DE MAHALANOBIS", subtitle = "Empresas eólicas") +
  ylab("MAHALANOBIS")

```

El gráfico de caja resultante será:



En el gráfico se observa que existen, por encima de la caja, varios *outliers*. Para identificarlos de modo concreto, hemos de calcular los cuartiles primero y tercero de la variable MAHALANOBIS y pasar el correspondiente filtro:

```
Q1M <- quantile (originales_maha$MAHALANOBIS, c(0.25))
Q3M <- quantile (originales_maha$MAHALANOBIS, c(0.75))
originales_maha %>% filter(MAHALANOBIS > Q3M + 1.5*IQR(MAHALANOBIS) |
MAHALANOBIS < Q1M - 1.5*IQR(MAHALANOBIS)) %>% select(MAHALANOBIS)
```

Tras ejecutar el código anterior, se obtiene el siguiente listado de casos que se comportan, según su *distancia de Mahalanobis* observada, como *outliers*:

	MAHALANOBIS
Parque Eolico La Boga SL.	23.68380
Guzman Energia SL	28.92177
Parque Eolico Santa Catalina SL	34.96254
WPD Wind Investment SL.	33.43117
Molinos Del Ebro SA	30.13740
Tarraco Eolica SA	29.79795
WPD Parque Eolico Las Panaderas SL.	18.83613
Eolica Navarra SL	50.37161
Parque Eolico El Moral SL	26.16517

Si, tras el estudio de los valores que toman las variables originales en estos casos, se decide eliminarlos, el código será:

```
muestra <- originales_maha %>% filter(MAHALANOBIS <= Q3M +  
1.5*IQR(MAHALANOBIS) & MAHALANOBIS >= Q1M - 1.5*IQR(MAHALANOBIS))  
muestra <- muestra %>% select(-MAHALANOBIS)
```

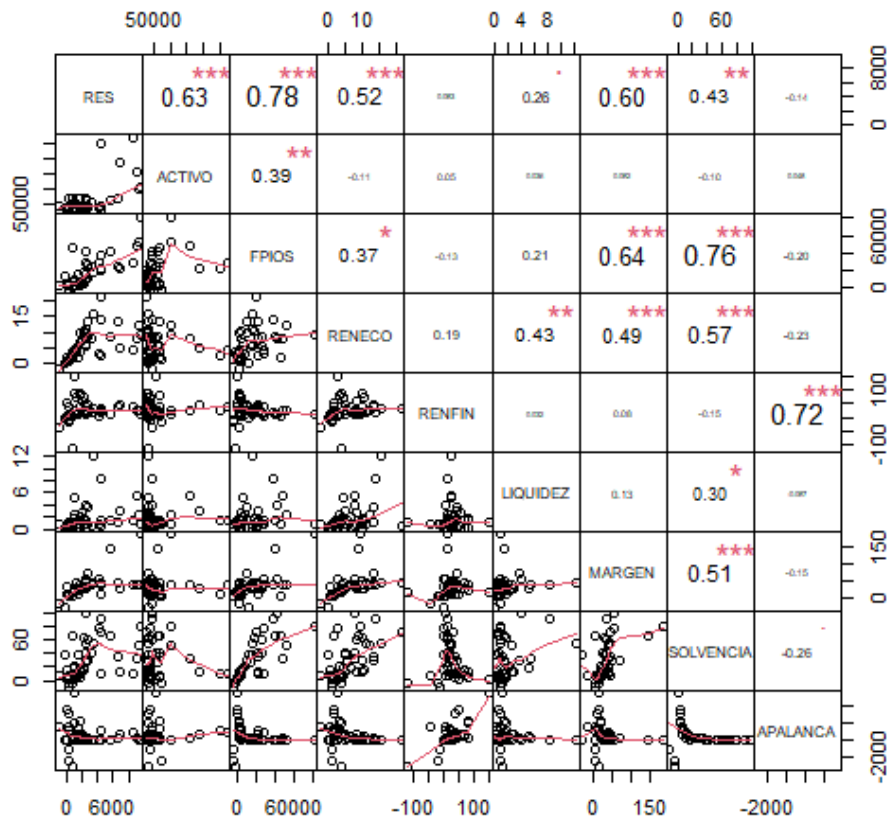
Se ha creado un nuevo *data frame* llamado “muestra” con los casos que no son *outliers* (y que no contienen *missing values*), y se ha eliminado la variable MAHALANOBIS, puesto que su única utilidad era la de localizar y filtrar los outliers. Con este *data frame* “muestra” es con el que se procederá al cálculo de las componentes principales.

Correlaciones y cálculo de componentes.

El análisis de componentes principales solo tiene sentido si las variables comparten “información redundante” sobre los casos (empresas eólicas). Un modo de comprobarlo es calcular las correlaciones entre las variables del estudio. Las correlaciones pueden obtenerse mediante la función `chart.Correlation()` del paquete `PerformanceAnalytics`:

```
library(PerformanceAnalytics)  
chart.Correlation(muestra, histogram = F, pch = 18)
```

El código anterior nos proporciona la siguiente matriz de correlaciones:



Puede apreciarse cómo existen altas correlaciones (en valor absoluto) entre algunas variables. Por tanto, tiene sentido hacer un análisis de componentes principales, ya que hay variables que parecen **compartir información**.

La obtención de las componentes se va a realizar mediante la función `prcomp()`. Es conveniente que activemos el argumento “scale” con “T” (true) para que las variables originales sean consideradas en sus **versiones tipificadas**. Vamos a asignar los resultados a un objeto de nombre, por ejemplo, “componentes”. La sintaxis es la siguiente:

```
# Obtención de componentes.
componentes <- prcomp (originales, scale=T)
summary (componentes)
```

Se obtienen los siguientes resultados:

```
Importance of components:
          PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8      PC9
Standard deviation  1.8892  1.3304  1.1952  0.9655  0.77718  0.63115  0.39059  0.32696  0.19562
Proportion of Variance 0.3966  0.1967  0.1587  0.1036  0.06711  0.04426  0.01695  0.01188  0.00425
Cumulative Proportion 0.3966  0.5932  0.7520  0.8555  0.92266  0.96692  0.98387  0.99575  1.00000
```


La “Standard deviation” es la raíz cuadrada de los autovalores asociados a cada componente. “Proportion of Variance” nos dice la proporción de la suma de varianzas de las variables originales (*comunalidad*) recogida por cada componente, proporción que se acumula en “Cumulative Proportion”. Nótese que las componentes aparecen ordenadas de más a menos importantes en función de la cantidad de varianza que capturan. En este caso, las cuatro primeras componentes acumulan más del 85% de la varianza (comportamiento) de las variables originales.

Los coeficientes o **cargas** de cada componente se obtienen pidiendo a nuestro objeto “componentes” el elemento “rotation”. Estas cargas las vamos a guardar en un nuevo objeto que llamaremos, por ejemplo, “cargas”:

```
# Cargas de cada componente.

Cargas <- componentes$rotation
round(cargas, 4)
```

Con el código anterior aparecerán, por columnas, las componentes; por filas, las variables originales; y en las intersecciones, los coeficientes (cargas) calculados (con un formato de 4 decimales). De este modo:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
RES	-0.4506	0.2151	-0.2741	0.0937	-0.2092	0.0963	-0.2840	0.4696	-0.5591
ACTIVO	-0.1613	0.2935	-0.6644	0.2925	-0.0500	0.1207	0.1577	-0.5535	0.1122
FPIOS	-0.4695	0.0191	-0.1986	-0.1873	0.3863	0.0409	0.0653	0.4048	0.6210
RENECO	-0.3688	0.0340	0.4372	0.1261	-0.4644	0.4159	-0.3209	-0.2228	0.3385
RENFIN	0.0369	0.6678	0.2867	-0.0694	-0.1542	0.1195	0.6289	0.1784	-0.0162
LIQUIDEZ	-0.2222	0.0194	0.3062	0.7923	0.2652	-0.3904	0.0690	0.0363	-0.0019
MARGEN	-0.4013	0.0772	0.0728	-0.4020	-0.2567	-0.7236	0.0164	-0.2757	-0.0184
SOLVENCIA	-0.4145	-0.1613	0.2364	-0.2111	0.4965	0.3289	0.2142	-0.3568	-0.4169
APALANCA	0.1760	0.6227	0.1252	-0.1266	0.4270	-0.0553	-0.5838	-0.1524	0.0145

Así, por ejemplo, la combinación lineal que define a la primera componente será:

$$y_{i1} = -0.4506 \cdot RES_i - 0.1613 \cdot ACTIVO_i - \dots + 0.1760 \cdot APALANCA_i$$

Además, se comprueba que las mayores cargas (en términos absolutos) de la primera componente son las correspondientes a FPIOS, RES, SOLVENCIA y MARGEN. En la segunda componente, las cargas correspondientes a RENFIN y APALANCA son las que tienen un mayor valor absoluto.

Como puede apreciarse, a menudo, encontrar un significado económico para las componentes no es una tarea sencilla, puesto que en realidad son elementos puramente matemáticos.

Retención de componentes.

Nosotros queremos menos componentes que variables originales, porque así simplificaremos la información suministrada por dichas variables. Esas componentes serán las **componentes principales**. Hay varios métodos de selección de componentes principales. Nos vamos a quedar con aquel que selecciona las varianzas de las componentes (o autovalores) mayores a uno. Para ello, primero calculamos los autovalores y los mostramos, y luego comprobaremos cuántos autovalores son mayores que 1.

Los autovalores, como ya vimos, son el cuadrado del elemento “Standard deviation” (sdev) del objeto “componentes” que hemos creado a partir de la función `prcomp()`. Hemos creado un *data frame* con estos autovalores calculados (y su orden, al que hemos llamado variable o columna “orden”, y que es un vector de números enteros consecutivos que va desde uno hasta número de variables originales o de componentes) y los hemos dispuesto en un gráfico de barras.:

```
# Determinación Componentes a retener.

# Criterio del Autovalor mayor que 1.

orden <- c(1:ncol(muestra))
autovalor <- componentes$sdev^2
autovalores <- data.frame(orden, autovalor)

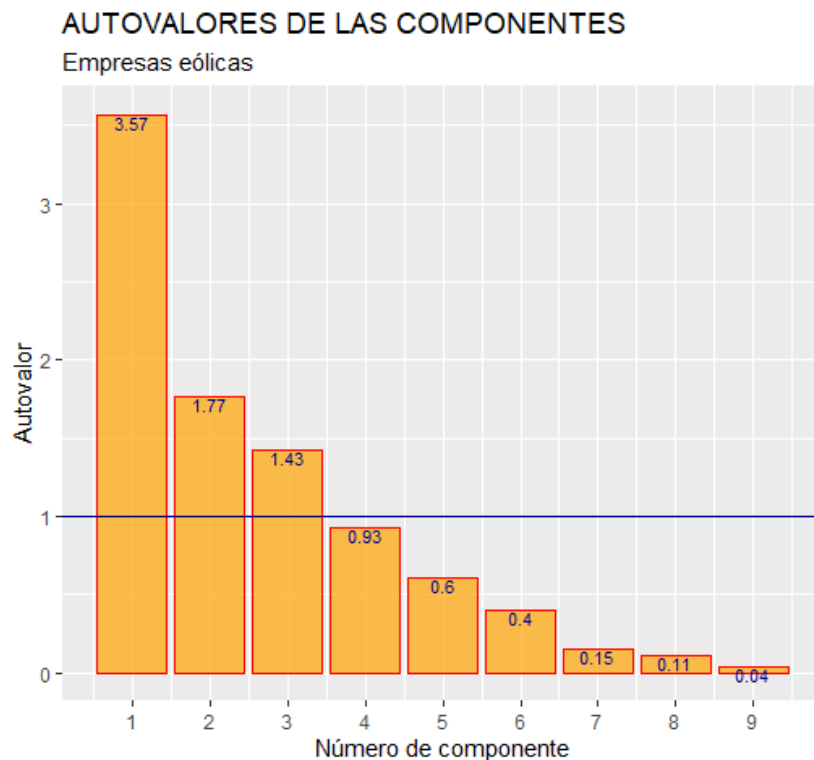
autograph <- ggplot(data = autovalores, map = (aes(x = orden, y =
autovalor))) +
  geom_bar(stat = "identity", colour = "red", fill =
"orange", alpha = 0.7) +
  scale_x_continuous(breaks=c(1:nrow(autovalores)))+
  geom_hline(yintercept = 1, colour = "dark blue") +
  geom_text(aes(label = round(autovalor,2)), vjust = 1,
colour = "dark blue", size = 3) +
  ggtitle("AUTOVALORES DE LAS COMPONENTES", subtitle =
"Empresas eólicas") +
  xlab("Número de componente") +
  ylab("Autovalor")

autograph
```

Respecto al gráfico, conviene recordar que, al ser un gráfico de barras, si no se quieren representar las frecuencias sino los valores que toma una variable (en este caso, “autovalor”) para cada valor de la otra variable (en

este caso, "orden"), en el *geom* habrá que añadir el argumento "stat" con el valor "identity". Además, se utiliza el elemento `scale_x_continuous()` para personalizar la escala del eje x, y que se divida dicho eje en tantos tramos como componentes hay.

En el gráfico obtenido se advierte que los 3 primeros autovalores son mayores que 1 (sobrepasan la línea horizontal), por lo que se retendrán las 3 primeras componentes, que serán las **componentes principales**:



Las tres primeras componentes, que son las componentes principales según este criterio de los autovalores superiores a 1, recogen el 75,20% de la varianza total de las variables originales o *comunalidad*. Esto ya se vio anteriormente al hacer `summary (componentes)`, aunque se puede representar gráficamente con el siguiente código.

```
Autovalores <- autovalores %>% mutate(variacum =
100*(cumsum((autovalor/nrow(autovalores))))
checkcp <- c(1:nrow(autovalores))
  for (I in 1:nrow(autovalores)) {
    if (autovalores$autovalor[i] >= 1) {
      checkcp[i] <- c("CP")
    } else {
      checkcp[i] <- c("NCP")}
  }
checkcp
```

Se comienza añadiendo al *data frame* “autovalores” una columna o variable que es la suma acumulada del porcentaje de *comunalidad* recogido por las sucesivas componentes, que están ordenadas de mayor a menor autovalor. Para calcular el porcentaje, se usa la función `cumsum()`, y se tiene en cuenta que, como las variables fueron tipificadas para calcular las componentes, la comunalidad, que coincide con la suma de las varianzas de las componentes (autovalores), es igual al número de variables o componentes (valor que toma la función `nrow()`).

Después, se ha creado un vector que contiene tantos elementos como variables o componentes hay en el análisis (vector “checkcp”), y se han determinado tales elementos a partir de un **bucle**, y con una **estructura condicional**.

El **bucle** comienza con la instrucción `for(){}`. En el paréntesis que sigue a la instrucción, se establece un valor “contador” (valor “i”, por ejemplo). Este valor va a ir aumentando de uno en uno, desde 1 hasta llegar al número de variables, componentes o autovalores (en nuestro caso, 9). Cuando “i” toma un valor, se realiza lo que está dentro de las llaves del bucle “{ }”. Luego vuelve a comenzar, tomando “i” el siguiente valor (1, 2, 3,...,9), hasta que termina.

Lo que hay dentro del bucle es una **estructura condicional** `if (){}/ else{}`. En el paréntesis que sucede a **if**, se establece una condición (en este caso, que el autovalor de la componente de orden “i” sea mayor o igual que 1). Si se cumple, el código a ejecutar será el que se encuentra entre las llaves inmediatamente posteriores al paréntesis de la condición (que el elemento “i” del vector “checkcp” tome valor “CP” (componente **principal**)). Si no se cumple la condición, se ejecutará el código situado dentro de las llaves localizadas después de la instrucción **else** (que el elemento “i” del vector “checkcp” tome valor “NCP” (**no** componente **principal**)). Así, se consigue que el vector “checkcp” distinga entre las componentes principales y las componentes que no serán retenidas, lo que se plasmará en el color de las barras del gráfico:

```
vacumgraph <- ggplot(data = autovalores, map = (aes(x = orden, y =
variacum, fill = checkcp))) +
  geom_bar(stat = "identity", colour = "red", alpha = 0.7)
+
  scale_x_continuous(breaks=c(1:nrow(autovalores)))+
  geom_text(aes(label = round(variacum,2)), vjust = 1,
colour = "dark blue", size = 3) +
  ggtitle("COMUNALIDAD ACUMULADA POR COMPONENTES", subtitle
= "Empresas eólicas") +
```

```

xlab ("Número de componente") +
ylab("Varianza acumulada")
vacumgraph

```

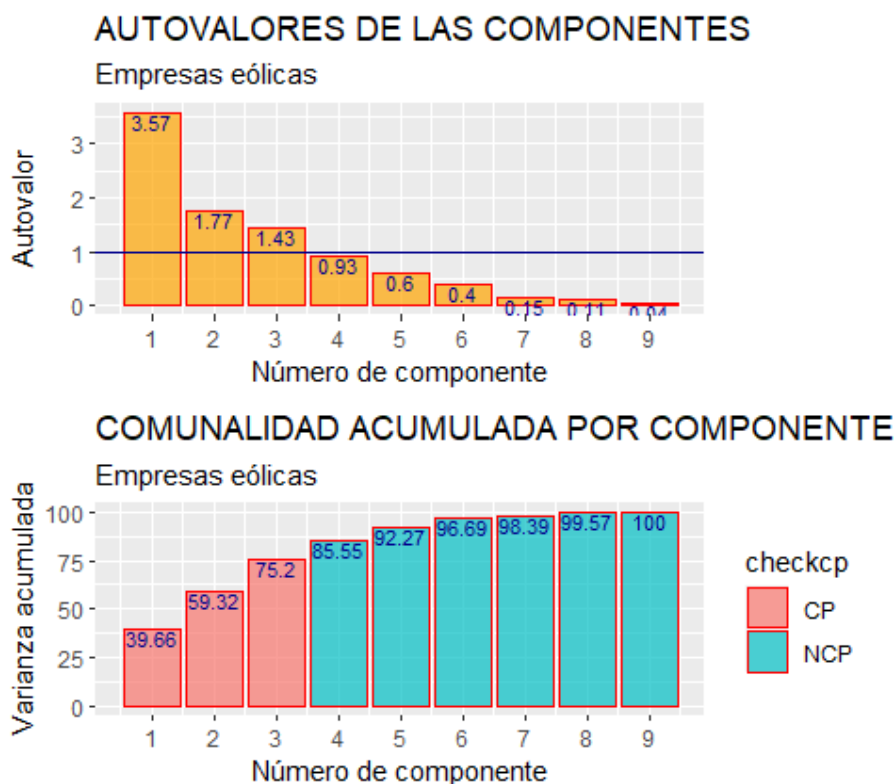
Posteriormente, mediante el paquete `patchwork()`, se han unido los dos gráficos anteriores, poniendo uno debajo del otro:

```

library (patchwork)
autograph / vacumgraph

```

El resultado obtenido es:



Se aprecia en el gráfico inferior cómo las tres primeras componentes, que son las principales, acumulan, en efecto, el 75,2% de la varianza total o *comunalidad*.

Así pues, para caracterizar el comportamiento de las empresas eólicas estudiadas en otros análisis (regresión, clúster, etc.) podrían emplearse como variables las puntuaciones para cada empresa de las tres primeras componentes (que serían las **componentes principales**), en lugar de utilizar los valores de las 10 variables originales.

Cálculo de las puntuaciones de los casos (scores).

Para obtener las puntuaciones de cada caso (empresa) en cada componente principal (que son las tres primeras componentes), simplemente debemos tener en cuenta que tales puntuaciones están guardadas en la matriz “x” del objeto `prcomp()` creado. Vamos a renombrar a las 3 primeras columnas (componentes) de esta matriz como “scores” y vamos a ver las puntuaciones de las primeras empresas:

```
# Scores.  
scores <- componentes$x[,1:3]
```

Así, por ejemplo, se obtienen, para las 6 primeras empresas de la muestra (excluidas empresas con *missing values* y outliers):

```
round(head(scores), 4)
```

Se han redondeado las puntuaciones a 4 decimales. Dichas puntuaciones son:

	PC1	PC2	PC3
Naturgy Wind, S.L.	-1.4408	2.0744	-3.5595
Al-Andalus Wind Power SL	-0.0498	1.4615	-3.0214
Acciona Eolica Del Levante SL	-0.8477	1.5180	-1.9849
Esquilvent SL	-2.7713	0.9430	-1.5715
Eolia Gregal De Inversiones SA.	-5.6488	0.4244	-1.1871
CYL Energia Eolica SL	-3.1226	0.4955	-0.7791

Estas puntuaciones se pueden obtener también, fácilmente, mediante cálculo matricial, ya que es el resultado de multiplicar la matriz de variables originales (tipificadas) por la matriz compuesta por las 3 primeras componentes, que son las componentes principales:

```
x <-data.matrix(scale(muestra))  
puntuaciones <- x %*% cargas[,1:3]  
puntuaciones
```

En el código anterior, se han tipificado los valores del *data frame* “muestra” mediante la instrucción `scale()`, y el resultado se ha pasado a formato de matriz, que hemos denominado “x”, para poder operar algebraicamente. Luego, se ha obtenido la matriz “puntuaciones” como multiplicación de la matriz anterior y la matriz de componentes principales (que se compone de

las 3 primeras columnas de las cargas o coeficientes calculados en el proceso de obtención de las componentes). Así, ejecutando ahora:

```
round(head(puntuaciones), 4)
```

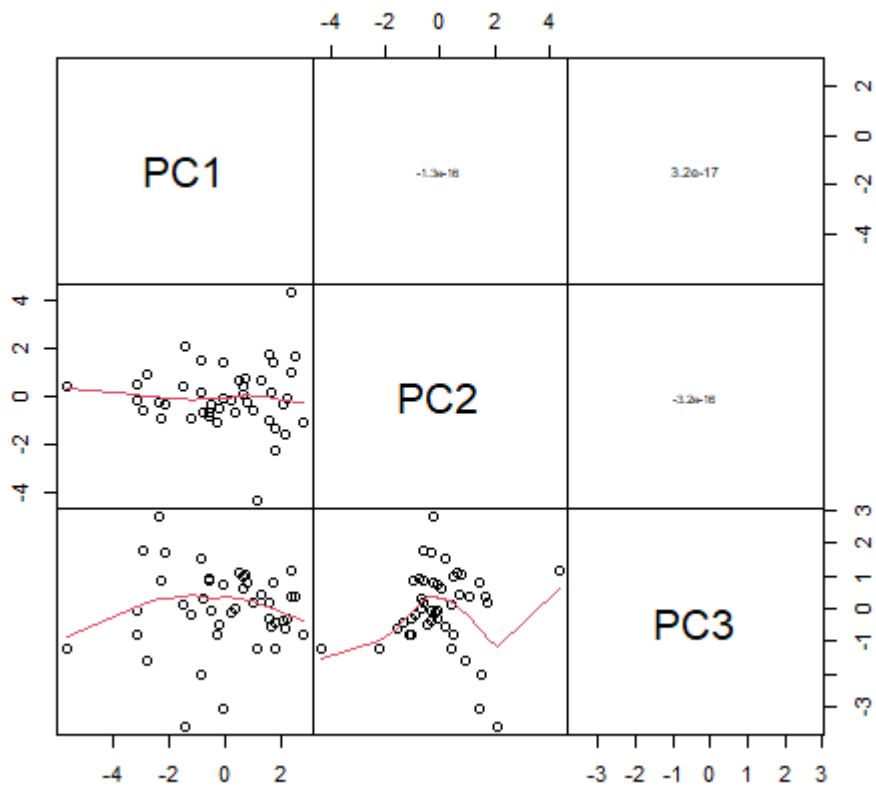
Se obtendrá el mismo resultado:

	PC1	PC2	PC3
Naturgy Wind, S.L.	-1.4408	2.0744	-3.5595
Al-Andalus Wind Power SL	-0.0498	1.4615	-3.0214
Acciona Eolica Del Levante SL	-0.8477	1.5180	-1.9849
Esquilvent SL	-2.7713	0.9430	-1.5715
Eolia Gregal De Inversiones SA.	-5.6488	0.4244	-1.1871
CYL Energia Eolica SL	-3.1226	0.4955	-0.7791


Las puntuaciones de los casos para cada una de las tres componentes principales pueden integrarse en un *data frame* y ser utilizadas como cualquier otra variable en un análisis multivariante, sabiendo que contienen gran parte de la información que, como caracterización de las distintas empresas, contenían las 9 variables originales.

Por último, cabe destacar que, si se calcula la matriz de correlaciones existentes entre las componentes principales:

```
chart.Correlation(puntuaciones, histogram = F, pch = 18)
```



Se obtiene que dichas correlaciones son prácticamente nulas.

This work © 2022 by [Miguel Ángel Tarancón](#) and [Consolación Quintana](#) is licensed under [Attribution-NonCommercial-NoDerivatives 4.0 International](#) 

Updated: 22/10/2022