Análisis descriptivo inicial.

Introducción.

Antes de la aplicación de técnicas complejas que permitan extraer de los datos conclusiones relevantes, conviene aplicar a tales datos unos **pasos iniciales** y **técnicas básicas** destinadas a conseguir dos objetivos:

- Preparar nuestros datos para que puedan ser procesados correctamente sin provocar distorsiones en los resultados.

- Obtener una visión inicial de la información básica que esconden nuestros datos, fundamentalmente en cuanto a las medidas básicas que caracterizan la distribución de frecuencias de las variables integradas en nuestros datos, así como, en el caso de contar con más de una variable, de las relaciones que existen entre ellas.

Además, es preciso tener en cuenta que, usualmente, es conveniente que estos rasgos iniciales que caracterizan a nuestra muestra o población sean plasmados **de un modo visualmente amigable, claro y conciso**.

En esta práctica, por medio de un ejemplo basado en información económico-financiera de una muestra constituida por 100 empresas dedicadas a la producción de electricidad mediante tecnología eólica, se mostrarán una serie de **buenas prácticas y análisis básicos** útiles a la hora de preparar y analizar inicialmente nuestro conjunto de datos.

Importando los datos.

Vamos a suponer que trabajamos dentro de un proyecto que hemos creado previamente, de nombre “explora”. Dentro de la carpeta del proyecto

guardaremos el *script* llamado "explora_describe.R", y el archivo de Microsoft® Excel® llamado "eolica_100_mv.xlsx". Si abrimos este último archivo, 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 las 100 empresas productoras de electricidad mediante generación eólica con mayor volumen de activo.

Luego cerraremos el archivo de Microsoft® Excel®, "eolica_100.xlsx", y volveremos a R-Studio. Después, abriremos nuestro script "explora_describe.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* (memoria) de objetos de anteriores sesiones de trabajo.

Para importar los datos que hay en la hoja "Datos" del archivo de Microsoft® Excel® llamado "eolica_100_mv.xlsx", ejecutaremos el código:

```
library(readxl)
eolica_100 <- read_excel("eolica_100_mv.xlsx", sheet = "Datos")
summary (eolica_100)
```

R ha considerado la primera columna como una variable de tipo cualitativo, atributo, o factor. En realidad, esta columna no es una variable, sino que está formada por los nombres de los diferentes casos u observaciones. Para evitar que R tome la columna de los nombres de los casos como una variable más, podemos redefinir nuestro *data frame* diciéndole que tome esa primera columna como el conjunto de los *nombres de los individuos*:

```
eolica_100 <- data.frame(eolica_100, row.names = 1)
```

Si hacemos ahora un summary():

```
summary (eolica_100)
```

Observaremos que ya no aparece NOMBRE, puesto que la columna correspondiente ya no es considerada como una variable.

Análisis de una sola variable. Buscando *missing values* y *outliers*.

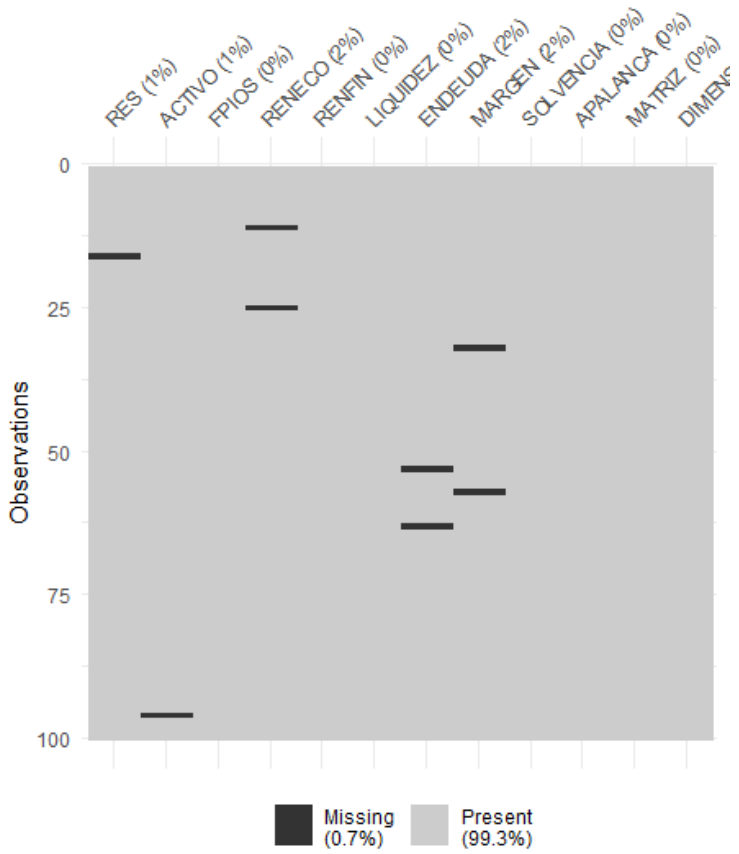
Vamos a suponer que la variable que queremos estudiar es la variable *Rentabilidad Económica* (RENECO).

La primera acción que debe realizarse es el comprobar que todos los casos (empresas) tienen su correspondiente dato o valor para la variable (RENECO), es decir, que no existen **valores perdidos o *missing values***.

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:

```
library(visdat)
vis_miss(eolica_100)
```

El resultado del código anterior es el siguiente gráfico:



Puede observarse que, en el caso concreto de la variable RENECO, un 2% de los casos no tienen dato (es decir, 2 casos de los 100). Para localizar los casos concretos, puede recurrirse a utilizar las herramientas de manejo de *data frames* del paquete dplyr. En concreto, realizaremos una **copia** del *data frame* original, "eolica_100", a la que llamaremos "muestra", que es con la que trabajaremos (para mantener la integridad del *data frame* original); y filtraremos los casos para **detectar aquellos que carecen de valor** en la variable "RENECO":

```
library (dplyr)
muestra<- select(eolica_100, everything())
muestra %>% filter(is.na(RENECO)) %>% select(RENECO)
```

La función is.na() comprueba si, en la posición correspondiente a una fila o caso, para la variable escrita en el argumento; hay o no un dato o valor. Como resultado se obtienen dos empresas, para las que se puede comprobar que no hay valor para la variable RENECO:

```
                      RENECO
Viesgo Renovables SL.     NA
Sargon Energias SLU       NA
```

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** de RENECO que no están disponibles, **o recurrir a alguna estimación** para los mismos y asignarlos. 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, vamos a suponer que hemos optado por esta última vía, al no conseguir unos valores más o menos verosímiles de RENECO para las empresas de las que se carece de dato. Esta eliminación de casos se podrá realizar mediante el código:

```
muestra <- muestra %>% filter(! is.na(RENECO))
```

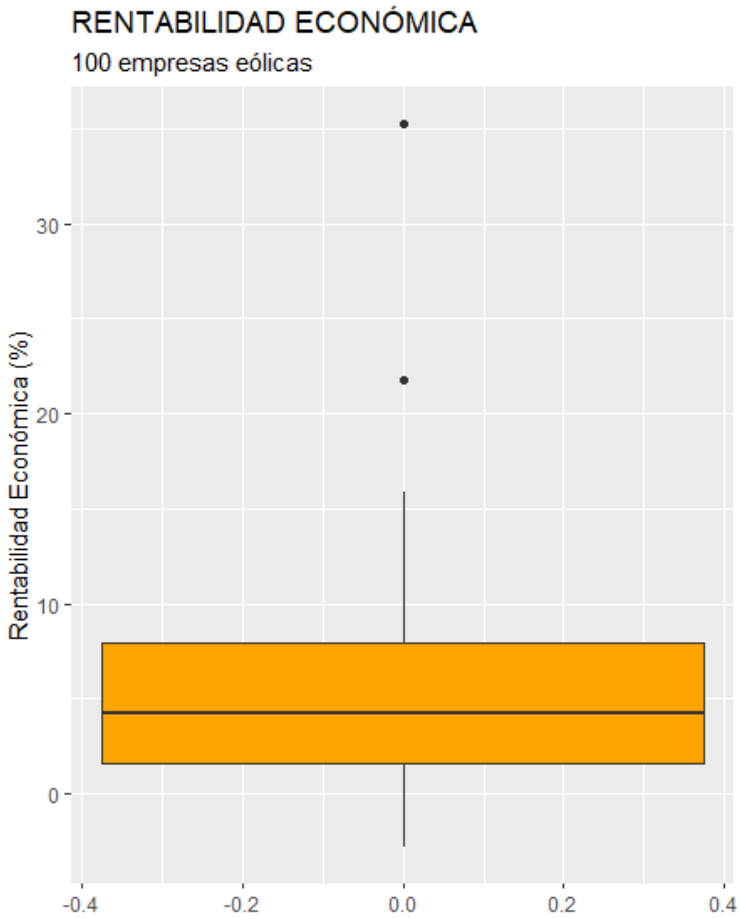
El operador "**!**" significa "no".

Podemos comprobar cómo en el *Global Environment* aparece el *data frame* "muestra" con dos casos menos (98).

Una vez tratados los casos con valores perdidos o *missing values*, **conviene detectar la posible presencia de *outliers*** o casos atípicos en la muestra, que pudieran desvirtuar los resultados derivados de ciertos análisis. Al trabajar con una sola variable métrica (la rentabilidad económica, RENECO), podemos intentar realizar esta tarea simplemente **representando gráficamente** la variable mediante un ***boxplot*** o gráfico de caja. Aplicaremos, por ejemplo, el código siguiente, que utiliza la gramática del paquete ggplot2:

```
library (ggplot2)
ggplot(data = muestra, map = (aes(y = RENECO))) +
    geom_boxplot(fill = "orange") +
    ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas eólicas")
+
    ylab("Rentabilidad Económica (%)")
```

Obteniéndose el gráfico:



Recordaremos que la caja contiene una línea horizontal que es la mediana. La caja contiene el 50% de los casos centrales (los que van del primer cuartil al tercero, cuya diferencia se llama *rango intercuartílico*), y contiene una línea horizontal que es la mediana. Por arriba sobresale una línea que llega al valor más grande que no llega a ser atípico; y por debajo de la caja lo mismo pero mínimo. Un valor atípico es el que se aleja más de 1.5 veces el rango intercuartílico (altura de la caja) del tercer cuartil, por arriba; o del primer cuartil, por abajo. Se registran mediante puntos.

En nuestro caso, el *boxplot* ratifica la existencia de dos casos atípicos. Para identificar esos dos casos concretos, podemos recurrir al paquete dplyr, y **establecer un filtro** con el siguiente código:

```
Q1 <- quantile (muestra$RENECO, c(0.25))
Q3 <- quantile (muestra$RENECO, c(0.75))
Q1; Q3
muestra %>% filter(RENECO > Q3 + 1.5*IQR(RENECO) | RENECO < Q1 -
1.5*IQR(RENECO)) %>% select(RENECO)
```

En el código anterior, las dos primeras filas calculan los cuartiles primero (Q1) y tercero (Q3) mediante la función quantile(). Es preciso tener en

cuenta que esta función calcula los percentiles. Luego se filtran, mediante la función de dplyr filter(), los *outliers*, calculados como aquellos casos con valores de RENECO mayores que Q3 más 1,5 veces el rango intercuartílico de la variable; o menores que Q1 menos 1,5 veces dicho rango intercuartílico. Para calcular el rango intercuartílico se recurre a la función IQR(). Finalmente, con select(), se muestran los casos en la consola de R-Studio.

Al ejecutar el código anterior, se visualizarán esas dos empresas atípicas:

```
                    RENECO
Molinos Del Ebro SA 35.262
Sierra De Selva SL  21.761
```

Como ocurría con los *missing values*, el tratamiento de los *outliers* depende de la información que se tenga, existiendo varias alternativas (corrección del dato, estimación, etc.) Si no se tiene información fiable, y los *outliers* no representan una gran proporción respecto al total de casos, puede optarse por su eliminación de la muestra. En este ejemplo, efectivamente, **eliminaremos estas dos empresas con comportamiento atípico** en la rentabilidad económica, a fin de que su presencia en la muestra no **distorsione los resultados en la aplicación posterior de ciertas técnicas** (por ejemplo, un ANOVA o un análisis de regresión). Podemos hacerlo creando un nuevo *data frame* a partir de "muestra"; pero sin esos dos casos. Ese nuevo *data frame* se llamará, por ejemplo, "**muestra_so**":

```
muestra_so <- muestra %>% filter(RENECO <= Q3 + 1.5*IQR(RENECO) & RENECO
>= Q1 - 1.5*IQR(RENECO))
```

Es importante observar que, en el código de filter(), las desigualdades deben cambiar, así como el operador "**|**" por el operador "**&**".

En el *Global Environment* podemos comprobar cómo el *data frame* "muestra_so" posee el mismo número de variables que el *data frame* "muestra"; pero con dos observaciones o casos menos (96).

Descripción de una variable.

Una vez se tiene preparada nuestra base de datos, con un tratamiento adecuado de los *missing values* y de los *outliers,* y **antes** de proceder a la aplicación de una técnica adecuada, según los objetivos perseguidos en el

estudio; suelen presentarse una serie de **medidas** descriptivas y **gráficos** básicos que proporcionan una **idea inicial de la estructura** del sector para la variable o variables analizadas. Nos referimos a medidas y/o gráficos de posición, dispersión y forma (asimetría y curtosis).

Para obtener de un modo rápido estas medidas, puede recurrirse a la función descr() del paquete summarytools. Por ejemplo:

```
# Descriptivos básicos

library (summarytools)
descr(muestra_so$RENECO,
      stats = c("mean", "sd", "min", "q1", "med", "q3", "max", "iqr",
"cv"),
      transpose = FALSE,
      style = "simple",
      justify = "center",
      headings = T)
```

El resultado mostrado en la consola será:

```
Descriptive Statistics
muestra_so$RENECO
N: 96

               RENECO
------------- --------
    Mean         4.94
   Std.Dev       4.31
     Min        -2.81
     Q1          1.42
   Median        4.14
     Q3          7.84
     Max        15.88
     IQR         6.40
     CV          0.87
```

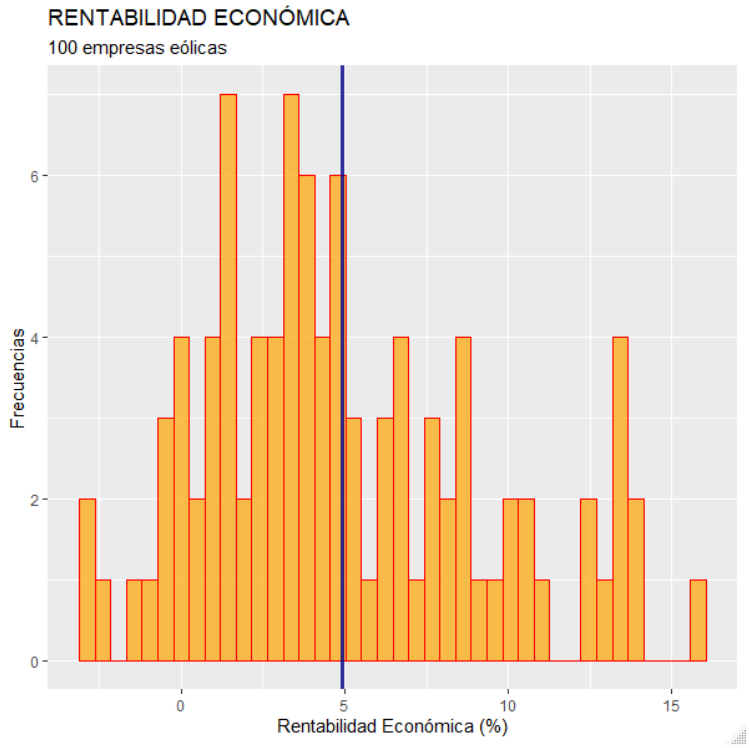
Estos valores se corresponden con el valor medio de RENECO, su desviación típica, valor mínimo, primer cuartil, mediana, tercer cuartil, valor máximo, rango intercuartílico, y coeficiente de variación o CV. Recordemos que este coeficiente mide el grado de apuntamiento o curtosis de la distribución, suponiendo que es acampanada y unimodal, de modo que un valor sensiblemente menor que cero indica que la distribución es “achatada” o *platicúrtica*, un valor sensiblemente mayor a uno evidencia una distribución muy “apuntada” o *leptocúrtica*, y un valor próximo a uno indica una distribución de apuntamiento *normal*, es decir, una distribución *mesocúrtica*.

No obstante, **el análisis gráfico suele dar una idea más atractiva de la estructura de la distribución de frecuencias en relación con la variable a analizar.**

Un gráfico fundamental es el **histograma** de la variable estudiada. Para ello, utilizaremos la gramática de ggplot2:

```
ggplot(data = muestra_so, map = aes(x = RENECO)) +
  geom_histogram(bins = 40, colour = "red", fill = "orange", alpha =
0.7) +
  geom_vline(xintercept = mean(muestra_so$RENECO), color = "dark blue",
size = 1.2, alpha = 0.8) +
  ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas eólicas")+
  xlab("Rentabilidad Económica (%)") +
  ylab("Frecuencias")
```

El gráfico correspondiente es:



En el gráfico vemos de un modo claro la distribución de frecuencias en cuanto a la rentabilidad económica (RENECO). Se ha incorporado una línea vertical axul para indicar la rentabilidad media. Entre otras cosas, se puede apreciar que la distribución de frecuencias es acampanada y *asimétrica positiva*.

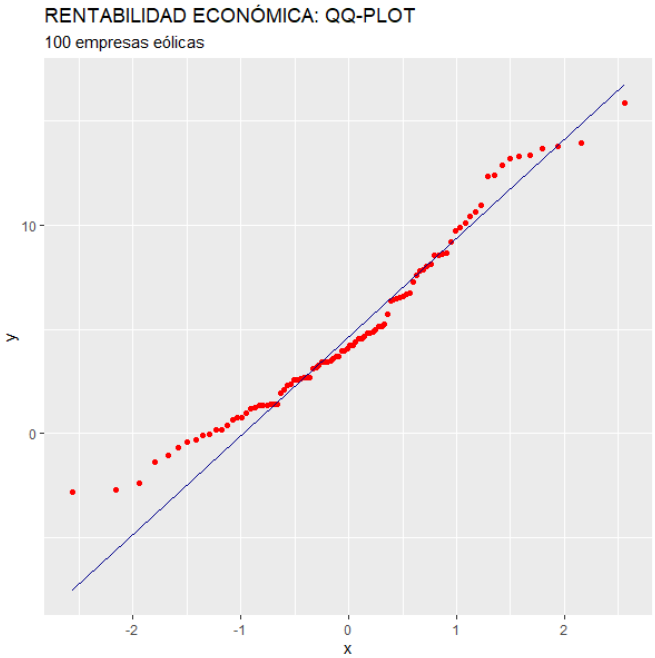
Normalidad.

En muchas técnicas multivariantes basadas en métodos inferenciales (por ejemplo, análisis de la varianza, o en la regresión lineal), se requiere que las variables sigan una **distribución normal**. Para comprobarlo, se puede a recurrir a análisis gráficos o formales, estos últimos basados en contrastes de normalidad.

Vamos a mostrar un **método gráfico** muy extendido. Comprobaremos la normalidad de la variable RENECO mediante un ***gráfico qq*** (cuantil-cuantil), que compara los cuantiles de nuestra muestra con los de una distribución normal teórica. Si los puntos se sitúan cercanos a la diagonal, entonces se asumirá un comportamiento (aproximadamente) normal:

```
ggplot(data = muestra_so, aes(sample = RENECO)) +
  stat_qq(colour = "red") +
  stat_qq_line(colour = "dark blue") +
  ggtitle("RENTABILIDAD ECONÓMICA: QQ-PLOT", subtitle = "100 empresas
eólicas")
```

El resultado de la ejecución del código anterior será el gráfico siguiente:



A veces, es difícil obtener una conclusión sólida con el gráfico *qq*; aunque en el ejemplo se aprecia, sobre todo en los primeros puntos, una separación notable de estos con respecto a la línea, lo que induce a pensar en que podría **no** seguirse una distribución normal.

Por el motivo anterior, si queremos ser más precisos, en lugar de un análisis gráfico se puede recurrir a realizar un análisis más formal, basado en la realización de contrastes de hipótesis. Una prueba muy usual es la **prueba de normalidad de *Shapiro y Wilk***, que tiene un buen comportamiento en muestras relativamente reducidas. En esta prueba, la hipótesis nula equivale al supuesto de normalidad. Para un 5% de significación estadística, un p-valor superior a 0.05 implicará el no-rechazo de la hipótesis de normalidad. Para realizar la prueba, se ejecutará el código:

```
shapiro.test(x = muestra_so$RENECO)
```

El resultado obtenido en la consola es:

```
      Shapiro-Wilk normality test

data:  muestra_so$RENECO
W = 0.9605, p-value = 0.005523
```

Como el p-valor es (muy) inferior a 0.05, se rechaza la hipótesis nula, lo que implica que, para una significación estadística del 5%, admitimos que RENECO **no** sigue, para nuestra muestra, un comportamiento normal, como ya se anticipó con el gráfico *qq*.

Análisis con dos variables. Buscando *missing values* y *outliers*.

Son muchas las técnicas aplicadas al análisis de la estructura de un sector basadas en una distribución de frecuencias bivariante (o multivariante). En este apartado nos centraremos en el caso de **variables métricas**, ya que al caso de atributos, variables categóricas o factores; le dedicaremos un tema en exclusiva. Algunas técnicas multivariantes son el análisis de componentes principales, el análisis de regresión, el análisis clúster…

Todos estos análisis requieren, de nuevo, de una fase inicial que ponga a punto la base de datos y ofrezca una fotografía inicial de cómo es el sector en cuanto a las variables en estudio. En este sentido, es conveniente aplicar, para cada variable, algunos de los análisis gráficos básicos vistos anteriormente.

A estos análisis básicos hay que incorporar, principalmente, algún estudio gráfico y alguna medida cuantitativa más, destinados fundamentalmente a

comprobar el **grado de intensidad en la relación estadística** entre las variables implicadas.

En nuestro ejemplo, vamos a incorporar al análisis la variable ACTIVO (*volumen de activos de la empresa en miles de euros*). Partiremos, como ya hicimos en el caso univariante, de la detección de valores perdidos o *missing values*. Para no modificar el *data frame* original ("eolica_100"), trabajaremos con una copia, llamada "muestra2":

```
muestra2 <- select(eolica_100, everything())
muestra2 %>% filter(is.na(RENECO) | is.na(ACTIVO)) %>% select(RENECO,
ACTIVO)
```

El operador "**|**" significa "o".

Con el código anterior se obtiene en la consola:

```
                            RENECO ACTIVO
Viesgo Renovables SL.           NA 269730
Sargon Energias SLU             NA  85745
La Caldera Energia Burgos SL 2.643     NA
```

Como ya se discutió anteriormente, si comprobamos que los *missing values* no pueden ser obtenidos o estimados mediante alguna otra vía, y son relativamente pocas observaciones, se podría optar por eliminarse. Vamos a suponer en este ejemplo, que ese es el caso:

```
muestra2 <- muestra2 %>% filter(! is.na(RENECO) & ! is.na(ACTIVO) )
```

El operador "**&**" significa "y".

El *data frame* "muestra2" contiene los mismos datos que "eolica_100", salvo los tres casos con *missing values* (97).

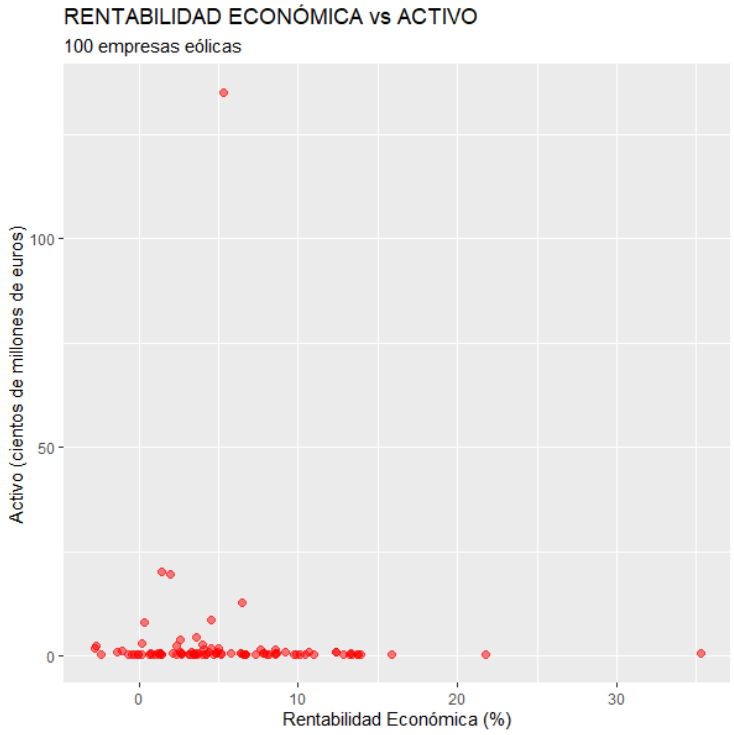
Se pasaría ahora a detectar los posibles *outliers*. Al trabajar con dos variables, una posibilidad cómoda para extraer una conclusión inicial es generar un gráfico o **diagrama de dispersión**. Utilizando la gramática de ggplot2:

```
# Localizando outliers

dispersion <- ggplot(data = muestra2, map = (aes(x = RENECO, y =
ACTIVO/100000))) +
              geom_point(colour = "red", size = 2, alpha = 0.5) +
```

```
              ggtitle("RENTABILIDAD ECONÓMICA vs ACTIVO", subtitle =
"100 empresas eólicas") +
              xlab("Rentabilidad Económica (%)") +
              ylab("Activo (cientos de millones de euros)")
dispersion
```

Puede apreciarse cómo en el "mapeo" de las coordenadas, el ACTIVO aparece divido entre 100000. Esto es simplemente para aminorar la escala del eje "y", y dotar de una escala más cómoda (visible) al gráfico. Eso ha hecho que en la etiqueta de este gráfico pongamos "cientos de millones de euros", en lugar de "miles de euros", que son las unidades de la variable ACTIVO. Por otro lado, el gráfico se ha asignado, primeramente, al objeto "dispersion", al que se ha llamado con posterioridad.

El código anterior da lugar al gráfico:

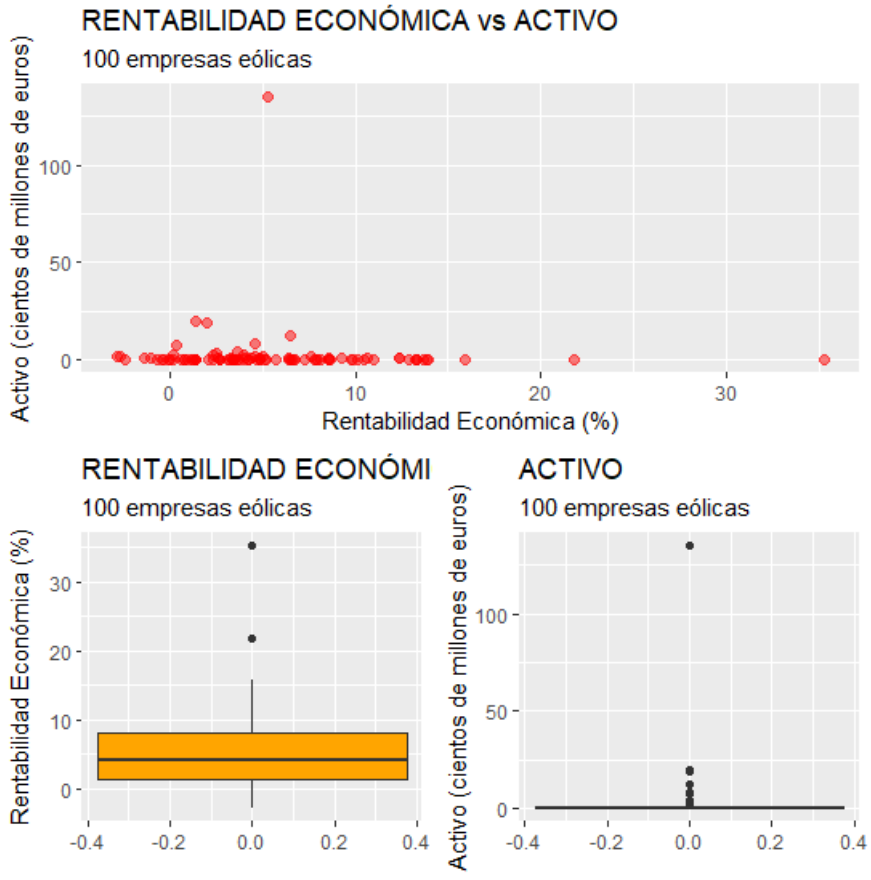


En el gráfico se distinguen varios puntos muy alejados del resto, que son candidatos a ser *outliers*.

Además, podríamos crear los diagramas de caja para cada una de las variables. Estos gráficos los vamos a asignar a dos objetos, "caja_RENECO" y "caja_ACTIVO". Posteriormente, con las facilidades del paquete patchwork se combinarán los tres gráficos, a fin de proporcionar una presentación más compacta. El código es el siguiente:

```
caja_RENECO <- ggplot(data = muestra2, map = (aes(y = RENECO))) +
               geom_boxplot(fill = "orange") +
               ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100
empresas eólicas") +
               ylab("Rentabilidad Económica (%)")

caja_ACTIVO <- ggplot(data = muestra2, map = (aes(y = ACTIVO/100000)))
+
               geom_boxplot(fill = "orange") +
               ggtitle("ACTIVO", subtitle = "100 empresas eólicas") +
               ylab("Activo (cientos de millones de euros)")

library (patchwork)
dispersion / (caja_RENECO | caja_ACTIVO)
```

La última línea de código, mediante el paquete patchwork, maqueta la visualización de los tres gráficos. El operador "/" indica que los gráficos siguientes se dispondrán inmediatamente debajo; mientras que "|" indica que el gráfico siguiente se dispone al lado del anterior. El resultado es:



Lo más destacable es, en el caso de la variable ACTIVO, cómo el diagrama de caja confirma la existencia de *outliers*, con un caso muy destacado.

Vamos a suponer que se decide eliminar los *outliers* de la muestra de empresas, a fin de evitar distorsiones en los resultados de la posterior aplicación de alguna técnica. El código para localizar los *outliers* será:

```
muestra2 %>% filter(RENECO > Q3_RENECO + 1.5*IQR(RENECO) | RENECO < Q1_RENECO - 1.5*IQR(RENECO) | ACTIVO > Q3_ACTIVO + 1.5*IQR(ACTIVO) | ACTIVO < Q1_ACTIVO - 1.5*IQR(ACTIVO)) %>% select(RENECO, ACTIVO)
```

Lo que produce en la consola el listado:

```
                              RENECO      ACTIVO
Holding De Negocios De GAS SL.  5.264 13492812.00
Global Power Generation SA.     1.393  2002458.00
Naturgy Renovables SLU          1.959  1956869.00
EDP Renovables España SLU       6.458  1275939.00
Corporacion Acciona Eolica SL   4.562   864606.00
Saeta Yield SA.                 0.360   796886.38
Elawan Energy SL.               3.615   443467.00
Olivento SL                     2.553   381206.98
Parque Eolico La Boga SL.       0.162   303904.36
Naturgy Wind, S.L.              3.949   273542.00
Al-Andalus Wind Power SL        2.349   249853.83
Innogy Spain SA.               -2.708   230338.51
Guzman Energia SL              -2.813   190286.98
Acciona Eolica Del Levante SL   4.985   188354.00
Biovent Energia SA              4.551   183899.00
Esquilvent SL                   7.621   157630.62
Molinos Del Ebro SA            35.262    62114.37
Sierra De Selva SL             21.761    27728.00
```

Son 18 empresas. Vamos a eliminarlas de la muestra, pero creando un nuevo *data frame*, "muestra2_so", para conservar el anterior:

```
muestra2_so <- muestra2 %>% filter(RENECO <= Q3_RENECO + 1.5*IQR(RENECO) & RENECO >= Q1_RENECO - 1.5*IQR(RENECO) & ACTIVO <= Q3_ACTIVO + 1.5*IQR(ACTIVO) & ACTIVO >= Q1_ACTIVO - 1.5*IQR(ACTIVO))
```

Si se construyen ahora los gráficos anteriores; pero con "muestra2_so", que ya no incluye *outliers*, se comprobará el cambio que experimentan tales gráficos:

```
dispersion_so <- ggplot(data = muestra2_so, map = (aes(x = RENECO, y = ACTIVO/100000))) +
  geom_point(colour = "red", size = 2, alpha = 0.5) +
  ggtitle("RENTABILIDAD ECONÓMICA vs ACTIVO", subtitle = "100 empresas eólicas") +
  xlab("Rentabilidad Económica (%)") +
  ylab("Activo (cientos de millones de euros)")

caja_RENECO_so <- ggplot(data = muestra2_so, map = (aes(y = RENECO))) +
  geom_boxplot(fill = "orange") +
  ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas eólicas") +
```

```
  ylab("Rentabilidad Económica (%)")

caja_ACTIVO_so <- ggplot(data = muestra2_so, map = (aes(y =
ACTIVO/100000))) +
  geom_boxplot(fill = "orange") +
  ggtitle("ACTIVO", subtitle = "100 empresas eólicas") +
  ylab("Activo (cientos de millones de euros)")

dispersion_so / (caja_RENECO_so | caja_ACTIVO_so)
```

Los gráficos son:

Aunque vuelven a aparecer dos *outliers* en la variable ACTIVO, en el gráfico de dispersión se aprecia una nube de puntos más homogénea que antes de haber eliminado los casos u observaciones anteriores.

Análisis con más de dos variables. Buscando *missing values* y *outliers*.

Si las variables que entran en nuestro análisis son más de dos (suponemos que todas están en escala métrica), la detección y, en su caso, eliminación de *missing values* será un proceso semejante al de los casos anteriores.

Por ejemplo, vamos a imaginar que queremos realizar un análisis en el que tendremos en cuenta las variables RENECO (rentabilidad económica), ACTIVO (volumen de activos de la empresa), MARGEN (margen de beneficio) y RES (resultado del ejercicio).

Para detectar los *missing values* procederemos creando una copia del *data frame* original (para preservarlo), llamada "muestra3" y ejecutando el código:

```
muestra3 <- select(eolica_100, everything())
muestra3 %>% filter(is.na(RENECO) | is.na(ACTIVO) | is.na(MARGEN) |
is.na(RES)) %>% select(RENECO, ACTIVO, MARGEN, RES)
```

El listado de casos con valores perdidos en estas variables será:

```
                           RENECO    ACTIVO   MARGEN       RES
Viesgo Renovables SL.          NA 269730.00   11.818  4609.000
Biovent Energia SA          4.551 183899.00   22.792        NA
Sargon Energias SLU            NA  85745.00 -615.625 -2216.000
Parc Eolic Sant Antoni SL   1.361  69654.00       NA   668.000
Eolica La Brujula SA        7.295  42146.98       NA  2306.062
La Caldera Energia Burgos SL 2.643        NA   14.448   511.304
```

Para eliminar estos casos (siempre que no se hayan podido obtener por otra vía o estimar) utilizaremos el código:

```
muestra3 <- muestra3 %>% filter(! is.na(RENECO) & ! is.na(ACTIVO) & !
is.na(MARGEN) & ! is.na(RES))
```

El *data frame* "muestra3" contiene los mismos datos que "eolica_100", salvo los 6 casos con *missing values* (94).

Para la detección de posibles *outliers*, al haber más de 2 variables, ya no puede utilizarse un gráfico de dispersión, porque implicaría más de dos ejes. Además, si las variables que entran en el análisis son numerosas, podría ser poco operativo estudiar las variables una a una. Una alternativa consiste en calcular la ***distancia de Mahalanobis*** de las variables del estudio, como **"resumen" del comportamiento de cada caso** en todas las variables del análisis. Así, primero vamos a calcular un vector con los valores de la *distancia de Mahalanobis* para las 4 variables en cada uno de los casos (empresas eólicas). Este vector lo denominaremos, por ejemplo, "muestra3.maha*".* Una vez calculado, lo añadiremos al *data frame* "muestra3" mediante la función de pegado de columnas, cbind():

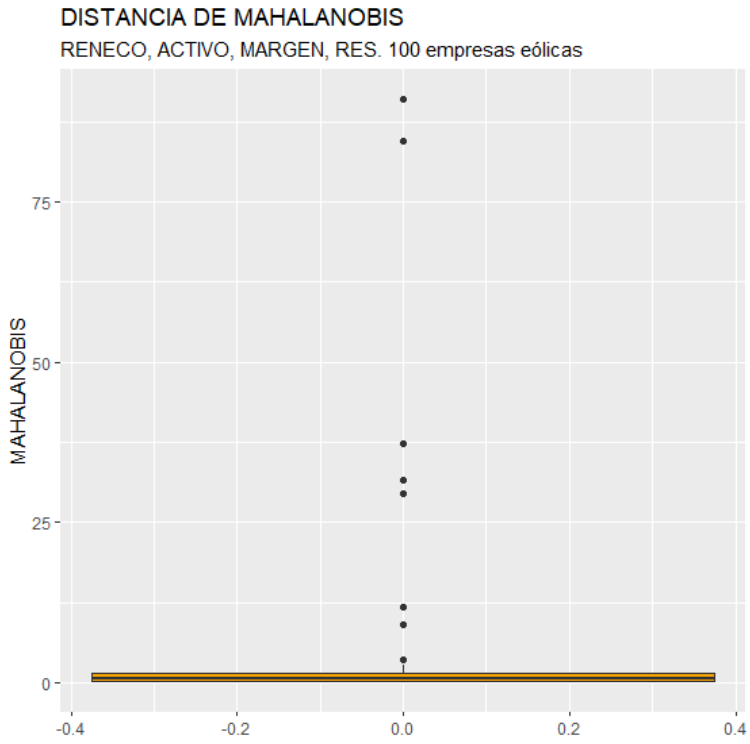
```
#Detectando y eliminando outliers.

muestra3.variables <- muestra3 %>% select(RENECO, ACTIVO, MARGEN, RES)
muestra3.maha <-mahalanobis(muestra3.variables,
                            center = colMeans(muestra3.variables),
                            cov = cov(muestra3.variables))
muestra3 <- cbind(muestra3, muestra3.maha)
```

Con el código anterior, se creó un *data frame* conteniendo solo las variables del análisis ("muestra3.variables"). Este *data frame* se utilizó para calcular la *distancia de Mahalanobis* teniendo en cuenta únicamente tales variables. Esas distancias se recogen en el vector "muestra3.maha", que pasa a integrarse en el *data frame* "muestra3" como una columna o variable más, mediante cbind().

Posteriormente, se puede construir el diagrama de caja de la variable muestra.maha, como cualquier otra variable. Antes, se ha cambiado el nombre de tal variable por MAHALANOBIS, mediante la función rename() de dplyr:

```
muestra3 <- rename(muestra3, MAHALANOBIS = muestra3.maha)
ggplot(data = muestra3, map = (aes(y = MAHALANOBIS))) +
    geom_boxplot(fill = "orange") +
    ggtitle("DISTANCIA DE MAHALANOBIS", subtitle = "RENECO, ACTIVO,
MARGEN, RES. 100 empresas eólicas ") +
    ylab("MAHALANOBIS")
```

El gráfico de caja resultante será:



Para saber de qué casos concretos se trata, se podrá ejecutar el código:

```
Q1M <- quantile (muestra3$MAHALANOBIS, c(0.25))
Q3M <- quantile (muestra3$MAHALANOBIS, c(0.75))
muestra3 %>% filter(MAHALANOBIS > Q3M + 1.5*IQR(MAHALANOBIS) |
MAHALANOBIS < Q1M - 1.5*IQR(MAHALANOBIS)) %>% select(MAHALANOBIS,
RENECO, ACTIVO, MARGEN, RES)
```

El resultado es:

```
                            MAHALANOBIS RENECO      ACTIVO    MARGEN         RES
Holding De Negocios De GASSL.   91.041690  5.264 13492812.00    91.152 727548.0000
Global Power Generation SA.     37.255573  1.393  2002458.00    22.403  39995.0000
Naturgy Renovables SLU          31.675561  1.959  1956869.00    20.442  42737.0000
Saeta Yield SA.                 11.891027  0.360   796886.38    16.258   2084.4760
Molinos Del Ebro SA             29.589696 35.262    62114.37    41.821  17026.2569
Tarraco Eolica SA                3.600426 12.868    38102.00   400.899   4953.0000
WPD Parque Eolico Navillas SL   84.589929 -0.416    35511.45 -2248.157   -110.9293
Brulles Eolica SL                3.599069 15.882    29722.58    47.227   3540.5693
Sierra De Selva SL               9.055155 21.761    27728.00    47.045   4525.0000
```

Si se opta por eliminar estos casos cara al análisis, se podrá crear un nuevo *data frame*, por ejemplo “muestra3_so”, con el código siguiente:

```
muestra3_so <- muestra3 %>% filter(MAHALANOBIS <= Q3M +
1.5*IQR(MAHALANOBIS) & MAHALANOBIS >= Q1M - 1.5*IQR(MAHALANOBIS))
```

El *data frame* “muestra3_so” será una réplica de “muestra3”, aunque sin incluir los casos detectados como atípicos (85 casos).

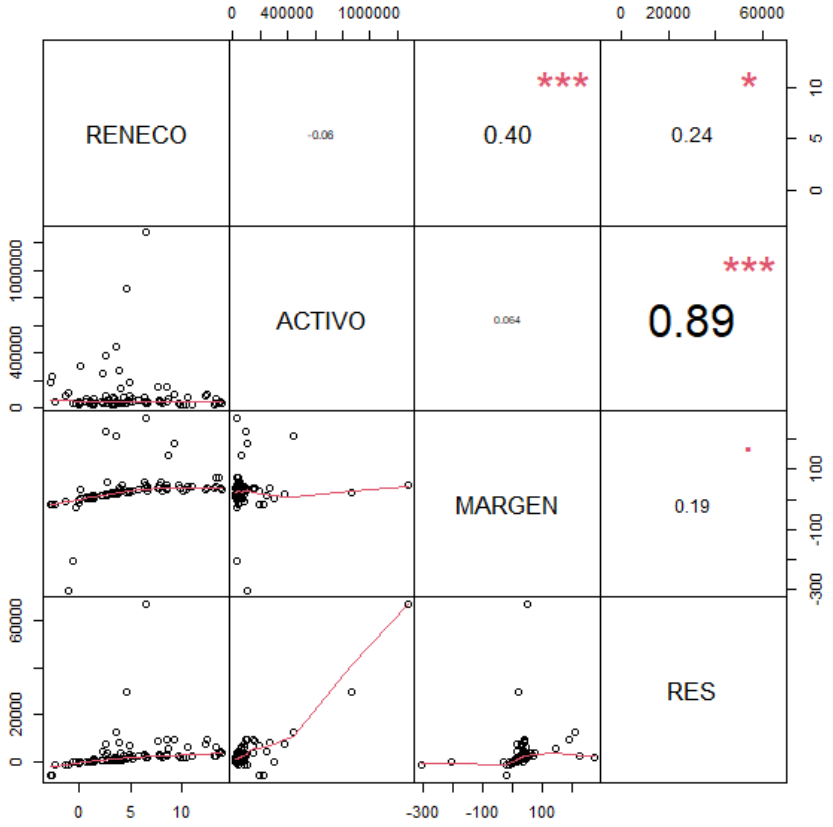
Análisis con más de dos variables. Correlación entre variables.

Cuando trabajamos con más de una variable, una característica muy importante viene dada por la intensidad con que tales variables están relacionadas entre sí, es decir, el estudio de las correlaciones. Un modo atractivo y rápido de visualizar la **matriz de correlaciones** de las variables es a través de la función chart.Correlation() del paquete PerformanceAnalytics:

```
# Correlaciones.

muestra3_so_variables <- muestra3_so %>% select(RENECO, ACTIVO, MARGEN,
RES)

library(PerformanceAnalytics)
chart.Correlation(muestra3_so_variables, histogram = F, pch = 18)
```

Previamente se ha creado el *data frame* "muestra3_so_variables" que contiene solo las variables del estudio.

El gráfico obtenido con el código anterior es:



Un coeficiente de correlación puede tomar un valor entre -1 (fuerte relación, en sentido opuesto) a 1 (fuerte relación, en el mismo sentido). Como puede apreciarse en el gráfico, las variables ACTIVO y RES mantienen una relación muy intensa y en sentido positivo. Entre MARGEN y RENECO existe también una relación de intensidad destacable. En cambio, ACTIVO y MARGEN; y RENECO y ACTIVO apenas están estadísticamente relacionadas.

Updated: 20/10/2022