



### **Tidyverse para gráficos: ggplot2.**

R, en su instalación básica, cuenta con funciones destinadas a crear gráficos y, de este modo, visualizar nuestros datos a fin de generar información y extraer conclusiones de un modo sencillo. No obstante, estas funciones, a veces, se quedan cortas, o requieren de un complejo y/o extenso código. Esta es la razón por la que en el **Tidyverse** se incluyó un paquete específico destinado a la construcción de gráficos de un modo flexible y amigable. Recordemos que el *Tidyverse* es un conjunto de paquetes / librerías con una filosofía común, como es el uso de ciertas estructuras gramaticales, que facilitan muchas de las tareas y análisis que podrían hacerse con el lenguaje R estándar.

Este paquete es **ggplot2**, que proporciona unas herramientas muy flexibles para visualizar gráficamente conjuntos de datos. A continuación, se expondrán los fundamentos de la **gramática** de **ggplot2** y se mostrará cómo construir algunos de los gráficos más habituales.

### **Obteniendo 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\_ggplot2.R” y el archivo de Microsoft® Excel® llamado “eolica\_100.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 total.

Luego cerraremos el archivo de Microsoft® Excel®, “eolica\_100.xlsx”, y volveremos a R-Studio. Después, abriremos nuestro *script* “explora\_ggplot2.R” con File → Open File... Este script contiene el programa que vamos a ir ejecutando en la práctica.

La primera línea / instrucción en los scripts suele ser:

```
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.xlsx”, ejecutaremos el código:

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

Podemos observar como en el *Environment* ya aparece un *data frame* que se llama “eolica\_100”, y contiene 11 columnas. 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)
```

En la línea anterior, hemos asignado al *data frame* “eolica\_100” los propios datos de “eolica\_100”; pero indicando que la primera columna no es una variable; sino que contiene el nombre de los casos. Si hacemos ahora un *summary()*:

```
summary(eolica_100)
```

Vemos que ya no aparece NOMBRE como variable y que, en el *Environment*, se recoge el *data frame* “eolica\_100” con 100 casos y con 10 variables.

## Cargando *ggplot2*.

A continuación, cargaremos el paquete *ggplot2*. Si nunca antes se ha utilizado este paquete, cuando lo intentemos activar con la función *library()* nos dará un error o nos dirá que previamente hay que instalarlo. En ese caso, iremos a la ventana inferior-derecha y pulsaremos la pestaña “*Packages*”, pulsaremos en “*Install*”, y emergerá una ventana donde dejaremos el “repositorio” que viene por defecto y, en el campo “*Packages*”, escribiremos el nombre del “paquete” (en nuestro caso *ggplot2*). Una vez descargado el “paquete”, podremos ejecutar el código sin problemas.:

```
#Cargando ggplot2  
  
library (ggplot2)
```

La instrucción principal del paquete *ggplot2* es *ggplot()*. A continuación, entre paréntesis, se deberán aportar una serie de argumentos o informaciones. Estas informaciones irán definiendo el gráfico en mayor o menor detalle.

En realidad, lo que se hace es definir el conjunto de datos a representar (que suelen estar contenidos en un *data frame*, o en varios), y a partir de ellos se van añadiendo capas gráficas (*geoms*) que son caracterizadas con ciertos atributos estéticos (*aesthetics*, o *aes*).

Vamos a generar algunos gráficos útiles en los siguientes apartados.

## Gráficos de una variable.

Comenzaremos generando algunos gráficos útiles para el análisis de una sola variable. Uno de estos gráficos, indispensable para tener una idea de la distribución de frecuencias que siguen los casos (en nuestro ejemplo, las empresas eólicas) con relación a una variable métrica es el **histograma**. Vamos a construir un histograma para la variable de rentabilidad económica, *RENECO*. El código será:

```
#Histograma  
ggplot(data = eolica_100, map = aes(x = RENEKO)) +  
  geom_histogram()
```

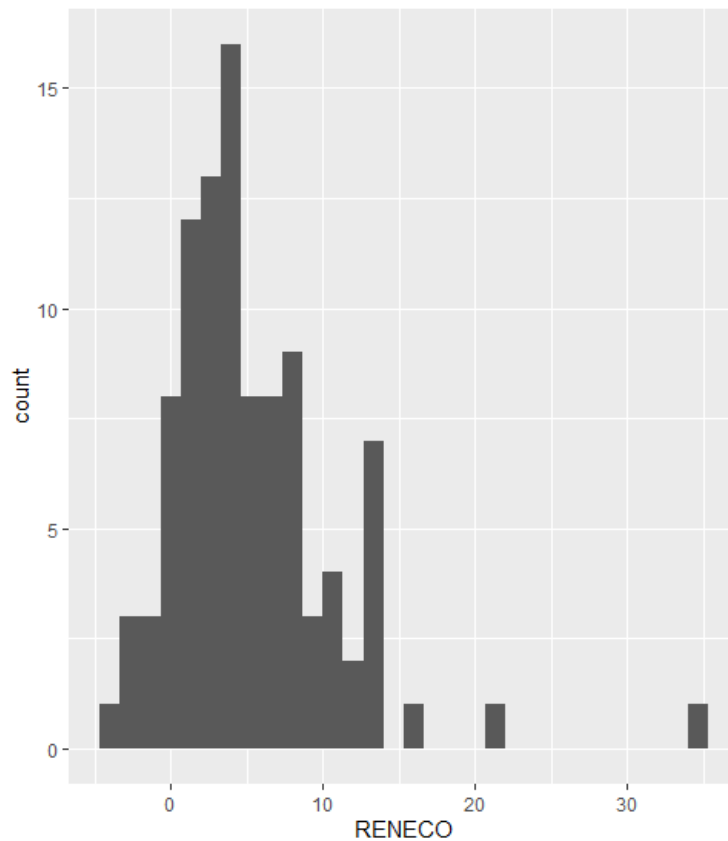
Como acabamos de decir, en primer lugar viene el comando `ggplot()`, seguido de unos paréntesis que recogen ciertas informaciones:

- **“data =”**, seguido de la fuente que almacena los datos a graficar (en nuestro caso, el *data frame* “eolica\_100”).
- **“map =”**, o “mapeo”, que define los aspectos del gráfico que dependen del valor de alguna o algunas variables. Siempre que alguna característica del gráfico no sea “fija”, sino que dependa de los valores que toma una variable, tal variable deberá ir indicada dentro de un *elemento estético* (*aes*). En el código de ejemplo, el elemento *aes* sirve para indicar que las coordenadas del eje x que toman los casos a representar, dependen de los valores de la variable RENEKO.

Para indicar que las siguientes líneas continúan con el código del gráfico, se añade al final de esta línea el símbolo “+”.

En la segunda línea, se establece el tipo de gráfico que se va a realizar, mediante la inclusión de un elemento **geom**. Para decir que lo que queremos construir es un histograma, el elemento *geom* será `geom_histogram()`.

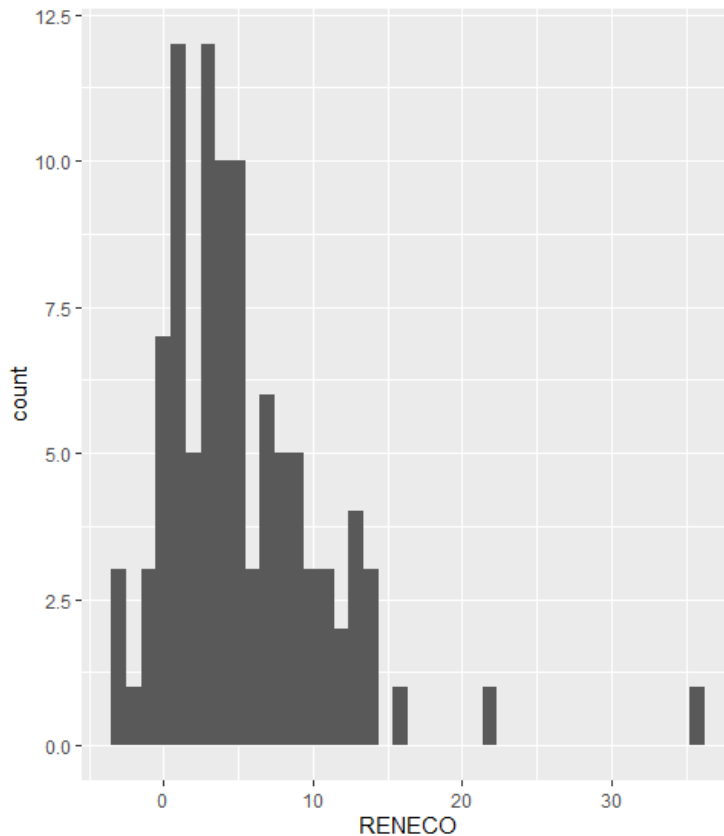
El resultado del código anterior es el siguiente gráfico:



Por supuesto, `ggplot2` permite personalizar y refinar la apariencia del gráfico. Uno de los aspectos que nos puede interesar modificar es el número de intervalos (barras), o *bins*. Por defecto, el número es 30. Para incrementar este número de barras a 40, por ejemplo; añadiremos en la línea del *geom* el argumento **“bins =”**:

```
ggplot(data = eolica_100, map = aes(x = RENECO)) +  
  geom_histogram(bins = 40)
```

El resultado es ahora:

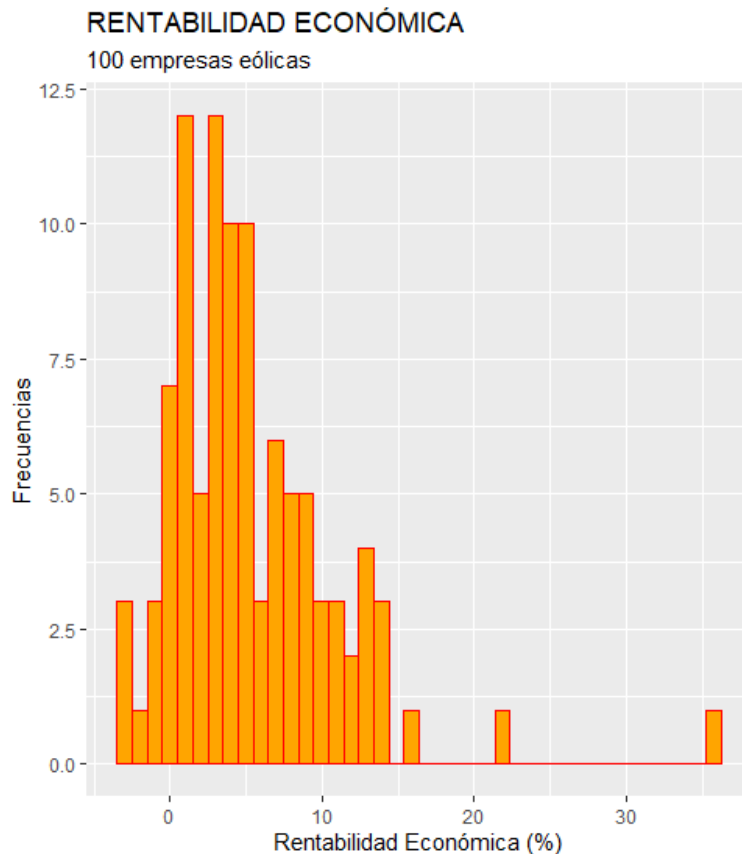


A continuación, vamos a modificar el color de las barras. Para el borde de estas, se utiliza el argumento **“colour =”**; y, para el relleno, **“fill =”**.

Por otro lado, vamos a mejorar la presentación del gráfico añadiéndole un título de figura, y unos títulos de gráfico. Hay que prestar atención en los signos **“+”** para que R entienda que el código de la siguiente línea pertenece al mismo gráfico que estamos diseñando:

```
ggplot(data = eolica_100, map = aes(x = RENECO)) +
  geom_histogram(bins = 40, colour = "red", fill = "orange") +
  ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas
eólicas")+
  xlab("Rentabilidad Económica (%)") +
  ylab("Frecuencias")
```

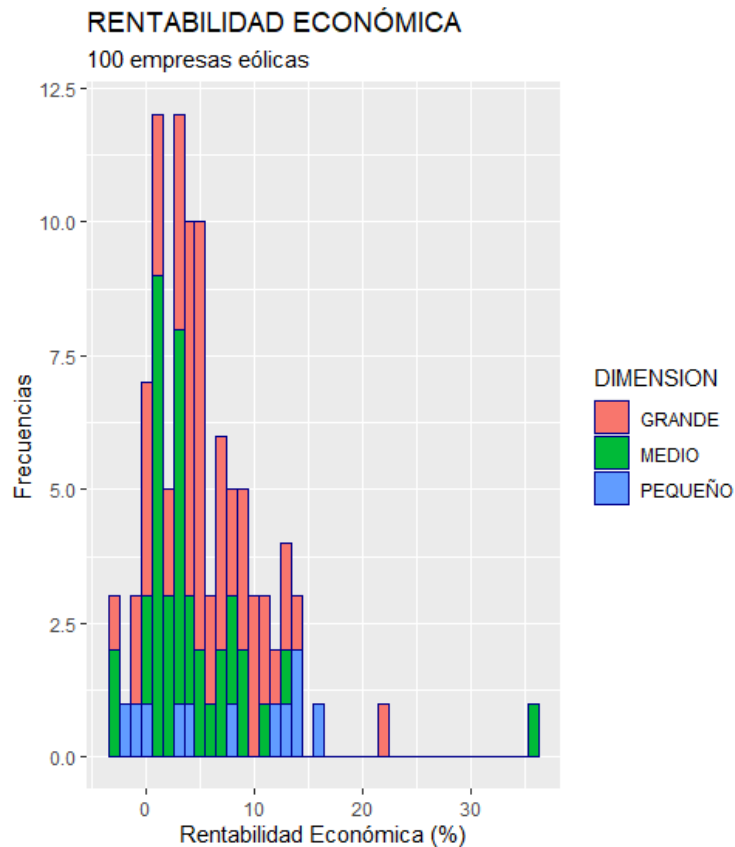
Con el código anterior, se obtiene el gráfico:



Ahora, refinaremos el gráfico teniendo en cuenta grupos de empresas. Entre las variables de nuestro *data frame* “eolica\_100”, existe una variable categórica, atributo o factor, denominada DIMENSION, que clasifica a las 100 empresas en “GRANDE”, “MEDIO” o “PEQUEÑO” según el volumen de los activos. Lo que vamos a hacer es crear, en el mismo gráfico, un histograma de la rentabilidad económica, pero para cada categoría de DIMENSION. Para ello, habrá que incluir esta variable categórica en el “mapeo”, en concreto mediante el argumento “fill =”. Es **necesario hacerlo dentro del elemento aes**, ya que el resultado (color de grupo de empresas) depende del valor que toma la variable DIMENSION para cada caso o empresa:

```
ggplot(data = eolica_100, map = aes(x = RENEKO, fill = DIMENSION)) +
  geom_histogram(bins = 60, colour = "dark blue") +
  ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas
eólicas")+
  xlab("Rentabilidad Económica (%)") +
  ylab("Frecuencias")
```

El resultado es el siguiente:



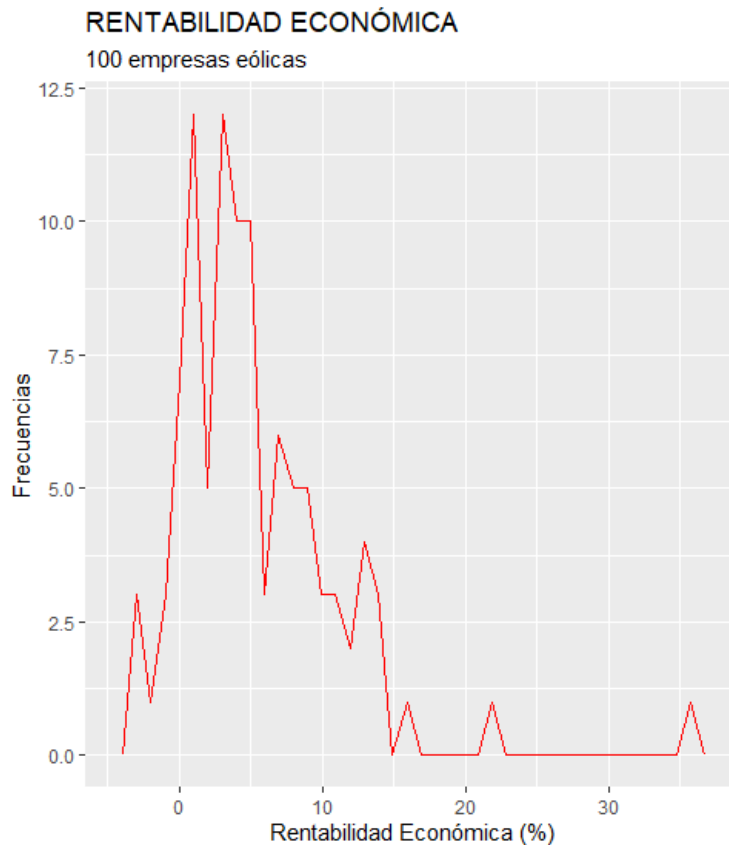
Como puede comprobarse, se superponen los tres histogramas, con tres colores diferentes, dependiendo de la dimensión considerada. Además, aparece, al lado derecho del gráfico, una leyenda que detalla qué color se asocia a cada uno de los grupos de empresas.

Los gráficos anteriores pueden transformarse fácilmente en **polígonos de frecuencias**, tan solo con algunos cambios en el *geom*. Así, por ejemplo, tenemos el siguiente código:

```
ggplot(data = eolica_100, map = aes(x = RENEKO)) +
  geom_freqpoly(bins = 40, colour = "red") +
  ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas
eólicas")+
  xlab("Rentabilidad Económica (%)") +
  ylab("Frecuencias")
```

El código anterior construye el siguiente polinomio de frecuencias:



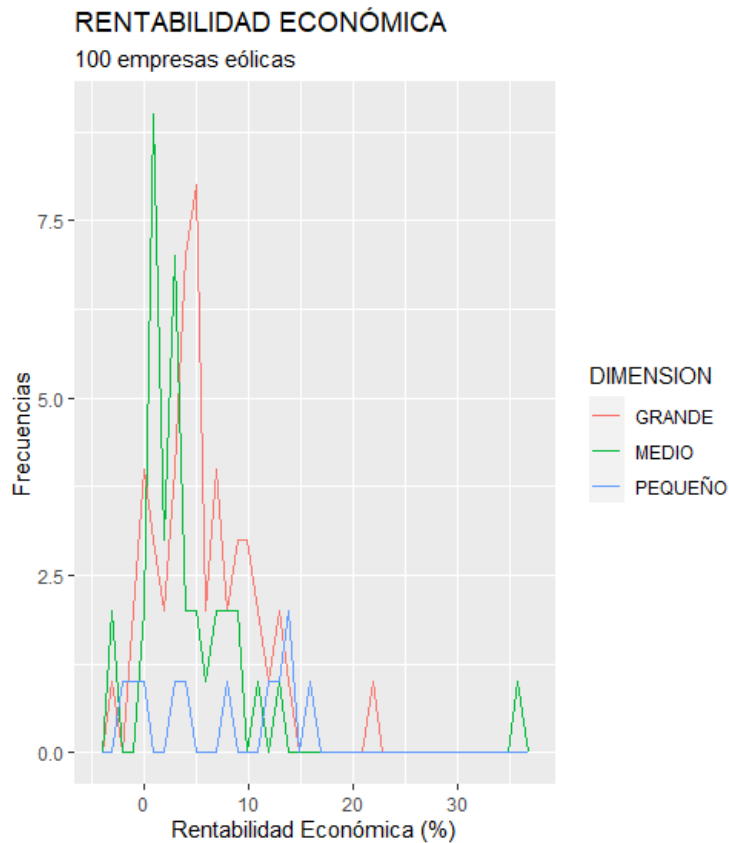


Si deseamos diferenciar entre grupos de empresas, una opción es ejecutar este otro código:

```
ggplot(data = eolica_100, map = aes(x = RENEKO, colour = DIMENSION)) +
  geom_freqpoly(bins = 40) +
  ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas
eólicas")+
  xlab("Rentabilidad Económica (%)") +
  ylab("Frecuencias")
```

En el código anterior, es interesante observar cómo la característica “colour =” ha pasado al bloque de “mapeo”, dentro del *aes*, ya que el color del polígono ya no es fijo; sino que depende del valor de la variable categórica, atributo o factor DIMENSION.

El resultado del código anterior es, en definitiva, el siguiente gráfico:

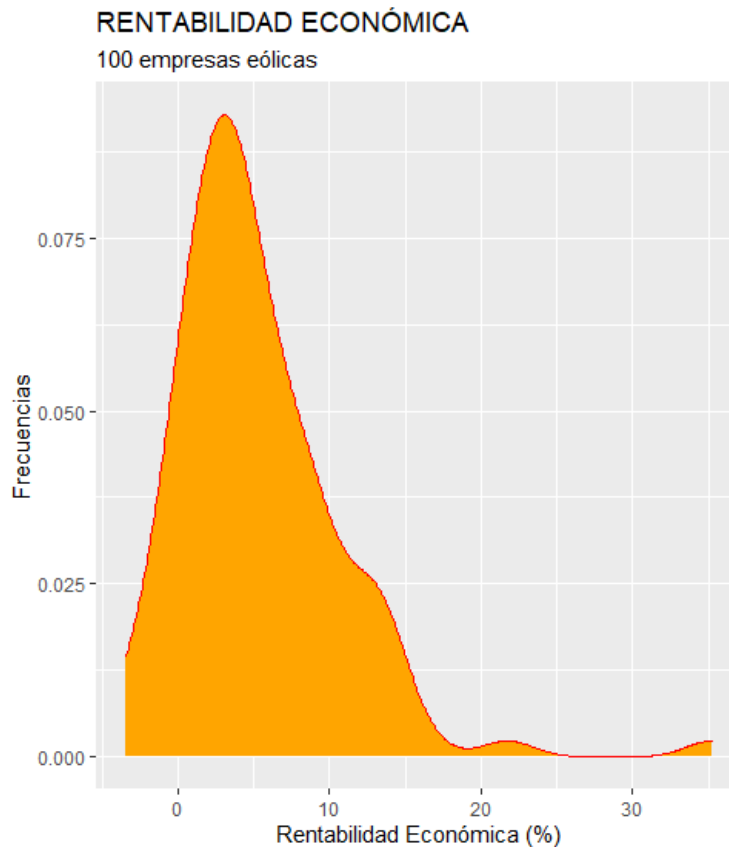


Finalmente, un gráfico parecido es el de **densidad**. Un gráfico de densidad estima la función de probabilidad de la variable representada. En realidad, podemos considerarlo como un histograma “suavizado”. Probemos a ejecutar este código:

```
ggplot(data = eolica_100, map = aes(x = RENEKO)) +
  geom_density(colour = "red", fill = "orange") +
  ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas
eólicas")+
  xlab("Rentabilidad Económica (%)") +
  ylab("Frecuencias")
```

En el código se observa la utilización del tipo de gráfico `geom_density()`. Además, desaparece el número de intervalos o *bins*, y se puede dotar a la función de densidad estimada de un color en su borde (“colour =”), y de un color de relleno (“fill =”).

Con todo ello, el resultado obtenido es el siguiente gráfico:

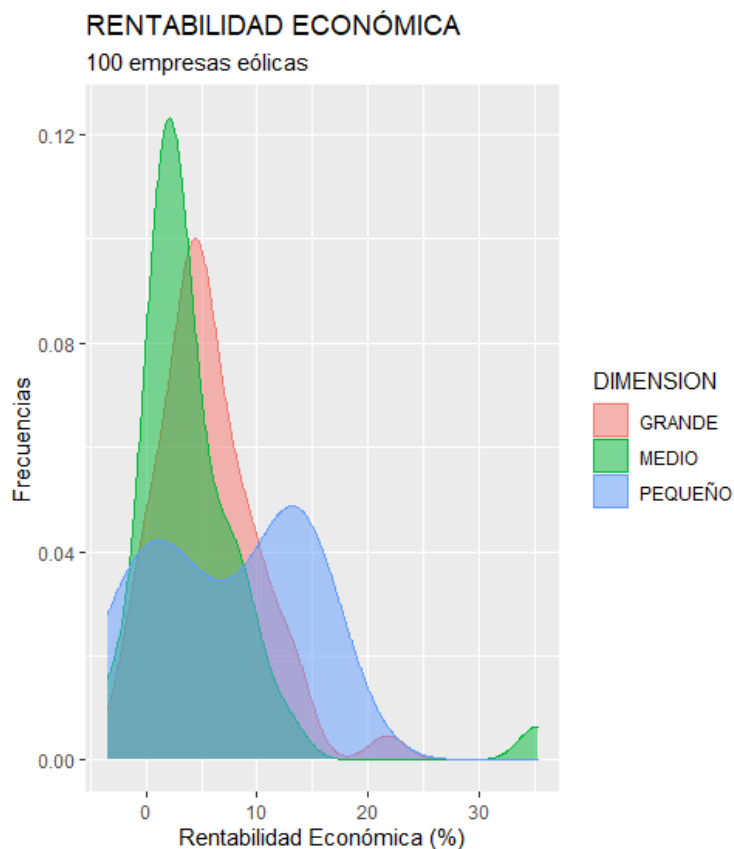


Como en casos anteriores, se puede crear una función de densidad estimada para cada grupo de empresas, eliminando las características, “colour =” y “fill =” del bloque del *geom*, y añadiéndolas en el “mapeo”, dentro del *aes*:

```
ggplot(data = eolica_100, map = aes(x = RENECO, colour = DIMENSION,
fill = DIMENSION)) +
  geom_density(alpha = 0.50) +
  ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas
eólicas")+
  xlab("Rentabilidad Económica (%)") +
  ylab("Frecuencias")
```

En efecto, los argumentos “colour =” y “fill =” han pasado a integrarse, en el “mapeo”, dentro de un *aes*, ya que van a variar dependiendo del grupo de pertenencia de la empresa (variable DIMENSION). Por otro lado, en el *geom* se ha añadido el argumento “alpha =”. Esta información consiste en un número de 0 a 1 que gradúa el grado de transparencia / opacidad de los rellenos de las figuras (en este caso las funciones de densidad estimadas) incluidas en los gráficos.

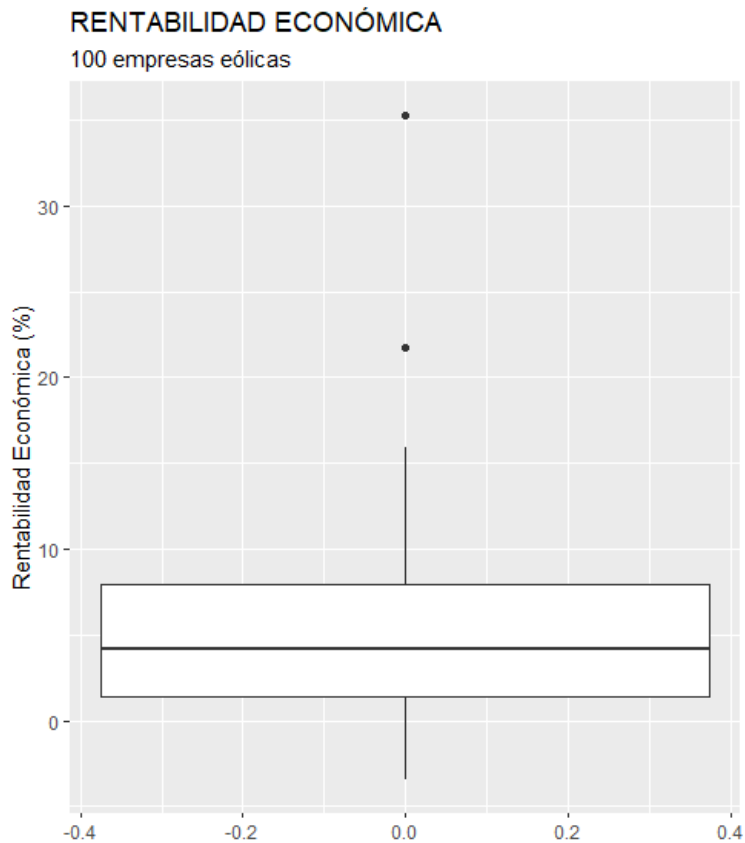
El resultado del código anterior es el siguiente gráfico:



Un tipo muy interesante de gráfico es el de “caja” (**box-plot**), que informa de la dispersión de una variable. Fijémonos en el siguiente código:

```
ggplot(data = eolica_100, map = (aes(y = RENECA))) +
  geom_boxplot() +
  ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas eólicas")
+
  ylab("Rentabilidad Económica (%)")
```

Puede observarse cómo en el “mapeo” se fija la variable que va a determinar las coordenadas del eje y. Como es una variable, hay que incluirla en el “mapeo” mediante una característica *aes*. El *geom* o tipo de gráfico es `geom_boxplot()`, y en este caso no le hemos añadido ninguna característica específica. Las últimas líneas configuran los títulos del gráfico y del eje “y”. El resultado de ejecutar el código es el siguiente gráfico:



El gráfico se caracteriza por una “caja” (rectángulo) central. Esta caja está limitada por el primer y tercer cuartil, luego recoge el 50% de los casos con una rentabilidad económica superior al 25% de los casos con menor rentabilidad, y por debajo del 25% de los casos con la rentabilidad más alta (“rango intercuartílico”). La caja, a su vez, está dividida en dos zonas por una línea que es la mediana de la distribución: la rentabilidad económica que divide a los casos en dos grupos con el mismo número de casos, uno con los casos de mayor rentabilidad, y otro con los casos de menor rentabilidad. Si esta línea (mediana) está en el centro de la caja, la distribución será simétrica, con bastante probabilidad.

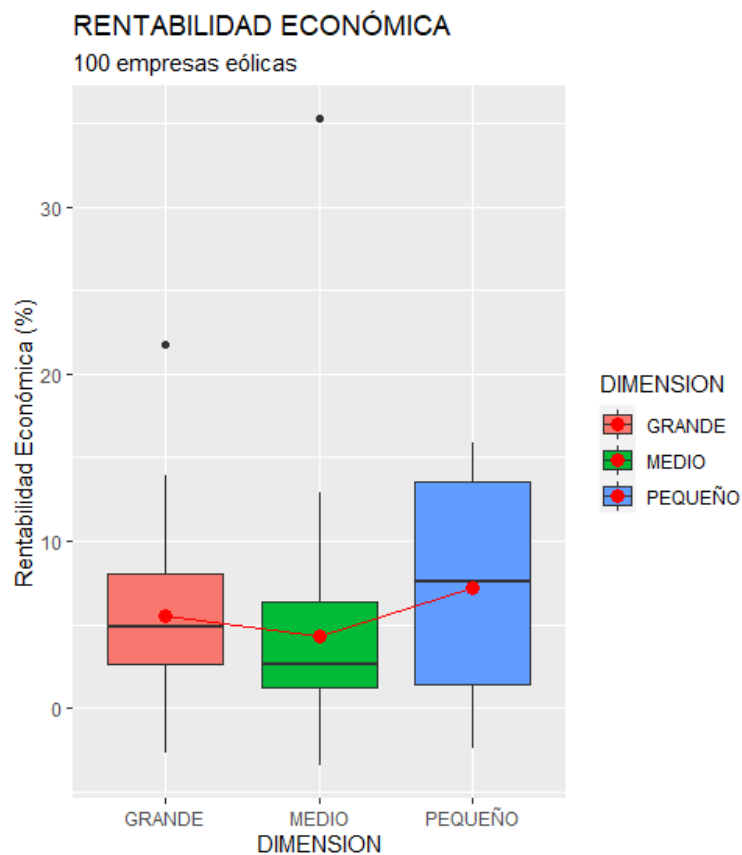
Por encima y por debajo de la caja se disponen dos líneas (llamadas “bigotes”). Estos “bigotes” recogen los valores inferiores al primer cuartil (comenzando por la base de la caja, hacia abajo), o superiores al tercer cuartil (comenzando por el techo de la caja, hacia arriba); y que están a menos de **1.5 veces** la altura de la caja. Los valores de rentabilidad inferiores al primer cuartil (por abajo) y superiores al tercero (por arriba), que están alejados de la caja en más de 1.5 veces de la altura de esta, se indican con puntos, y se corresponden con los casos conocidos como **casos atípicos** o **outliers**. La identificación de los *outliers* es una fase muy importante a la hora de aplicar algunas técnicas estadísticas.

En esta práctica, comprobamos cómo, en el caso de la rentabilidad económica (RENECO), existen dos *outliers*, es decir, dos casos que presentan una rentabilidad anormalmente elevada (más de un 20%).

Vamos a refinar el *box-plot* anterior. Por ejemplo, quizá nos pueda interesar Crear un *box-plot* para cada grupo de empresas, según su volumen de activos (DIMENSION). Esto lo conseguiremos con este código:

```
ggplot(data = eolica_100, map = (aes(x = DIMENSION, y = RENECO, fill = DIMENSION))) +  
  geom_boxplot() +  
  stat_summary(fun = "mean",  
              geom = "point",  
              size = 3,  
              col = "red") +  
  stat_summary(fun = "mean",  
              geom = "line",  
              col = "red",  
              map = (aes(group = TRUE))) +  
  ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas eólicas") +  
  ylab("Rentabilidad Económica (%)")
```

El gráfico obtenido es el siguiente:



Para construir una caja por grupo de empresas (según el volumen de activos), se ha incluido, en el “mapeo” de la primera línea, el eje x con la variable DIMENSION. Como, además, queremos que cada caja sea de un color diferente, hemos hecho que los colores de estas dependan de la variable DIMENSION; añadiendo en el *aes* del “mapeo” la característica “fill =” (que se refiere al color de relleno de las cajas).

Dentro de las cajas, se representan las medianas de la variable estudiada (RENECO), para cada grupo de empresas. Pero, a veces, es muy útil representar también otra medida de posición central como es la **media**. En los gráficos diseñados con *ggplot2* se pueden añadir medidas estadísticas mediante elementos *stat\_summary()*, algo parecido al *summarise()* de *dplyr*. Estos elementos contienen una serie de informaciones, como son “fun =” (la función, medida, o estadístico a representar), un *geom* (tipo de gráfico), y otros elementos opcionales. En el ejemplo, el primer bloque de *stat\_summary()* consigue puntear, para cada grupo de empresas, la media de RENECO en dicho grupo, en color rojo. Para comparar mejor estas medias, se ha procedido a unir los puntos con unos segmentos o líneas rojas, lo que se consigue con el segundo bloque de *stat\_summary()*. La última línea de ese bloque, `map = (aes(group = TRUE))`, obliga a que las líneas vayan de una media a otra de los grupos (de punto rojo a punto rojo).

Como última extensión, se ha considerado que, a veces, es conveniente tener en cuenta la posición de cada caso individual dentro del gráfico. Una opción es utilizar una capa o bloque *geom\_jitter()*. Con este *geom* se dispondrán, para cada grupo, los valores individuales de la variable RENECO; y para que estos, en su caso, no se solapen, se situarán un poco más a la izquierda o a la derecha, de modo aleatorio. Como los *outliers* son ya casos individuales, para que no se dupliquen con los provenientes del “jitter”, se indicará en el *geom\_boxplot()* que, en ese bloque gráfico, no se señalen los *outliers*. Esto se conseguirá con la instrucción `outlier.shape = NA`. El código, en definitiva, será:

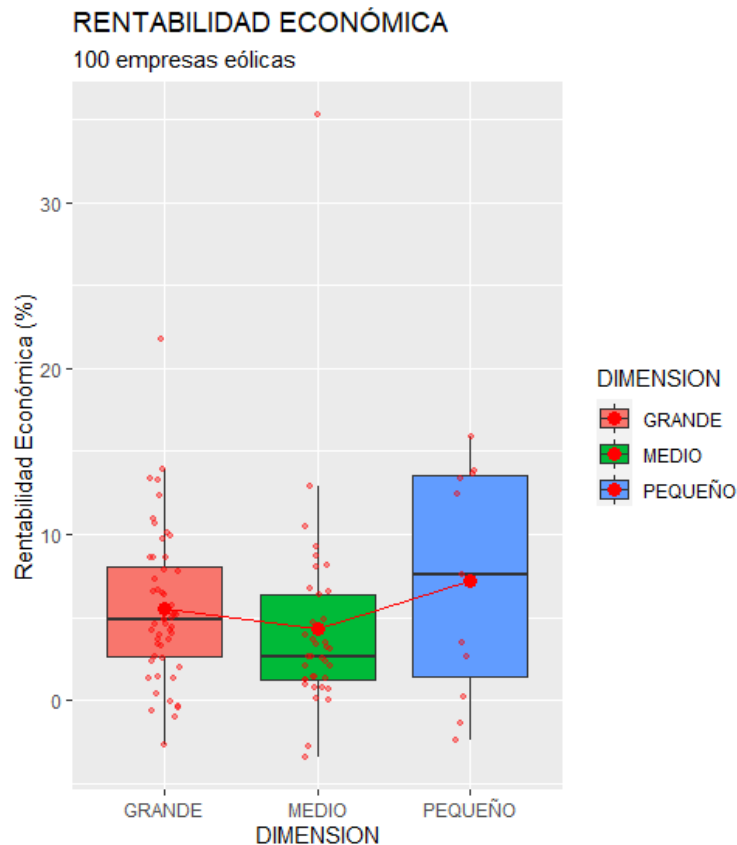
```
ggplot(data = eolica_100, map = (aes(x = DIMENSION, y = RENECO, fill = DIMENSION))) +
  geom_boxplot(outlier.shape = NA) +
  stat_summary(fun = "mean",
              geom = "point",
              size = 3,
              col = "red") +
  stat_summary(fun = "mean",
              geom = "line",
```

```

    col = "red",
    map = (aes(group = TRUE))) +
geom_jitter(width = 0.1,
            size = 1,
            col = "red",
            alpha = 0.40) +
ggtitle("RENTABILIDAD ECONÓMICA", subtitle = "100 empresas eólicas")
+
ylab("Rentabilidad Económica (%)")

```

El resultado de ejecutar el código anterior será el gráfico:



Como puede observarse, el `geom_jitter()` proporciona, en cada caja, la nube de casos (empresas) individuales, en cuanto a la rentabilidad económica (incluidos los *outliers*). Las características de estos puntos (amplitud del desplazamiento lateral “aleatorio”, tamaño, color, opacidad) se controlan con diversos argumentos (“width =”, “size =”, “col =”, “alpha =”).

### Gráficos de dispersión.

Pasamos ahora a comentar un tipo de gráfico muy común, cuando trabajamos con dos variables métricas: los **gráficos de dispersión** (o

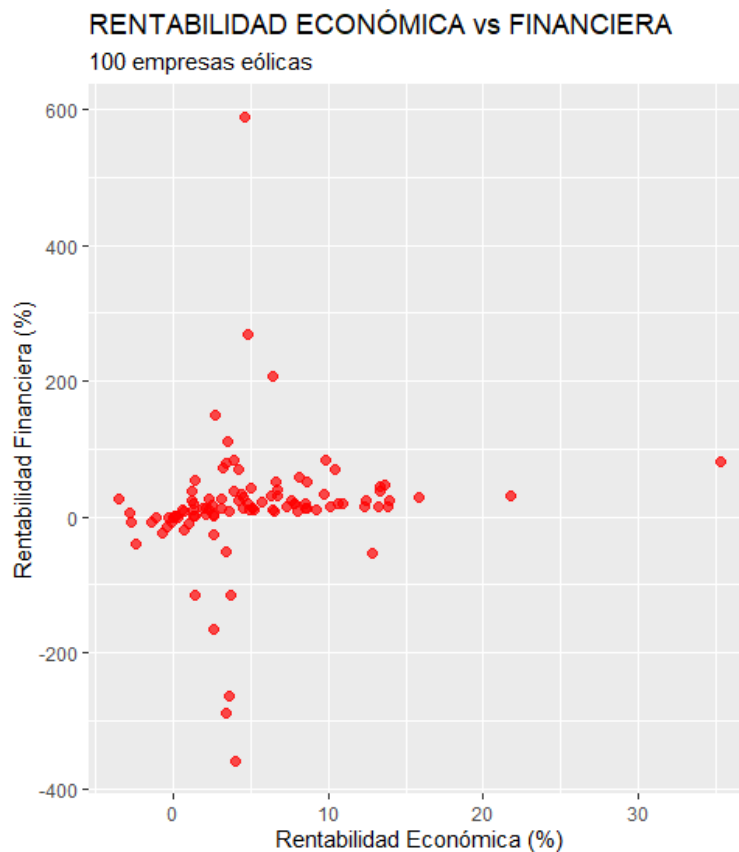


**scatterplots**). En este tipo de gráficos, cada variable ocupa un eje (x o y), y los puntos internos al gráfico representan los diversos casos u observaciones.

Como ejemplo, vamos a crear un diagrama de dispersión que represente las empresas eólicas en función de su rentabilidad económica (RENECO) y de su rentabilidad financiera (RENFIN). El código es el siguiente:

```
ggplot(data = eolica_100, map = (aes(x = RENECO, y = RENFIN))) +  
  geom_point(color = "red", size = 2, alpha = 0.7) +  
  ggtitle("RENTABILIDAD ECONÓMICA vs RENTABILIDAD FINANCIERA",  
  subtitle = "100 empresas eólicas") +  
  xlab("Rentabilidad Económica (%)") +  
  ylab("Rentabilidad Financiera (%)")
```

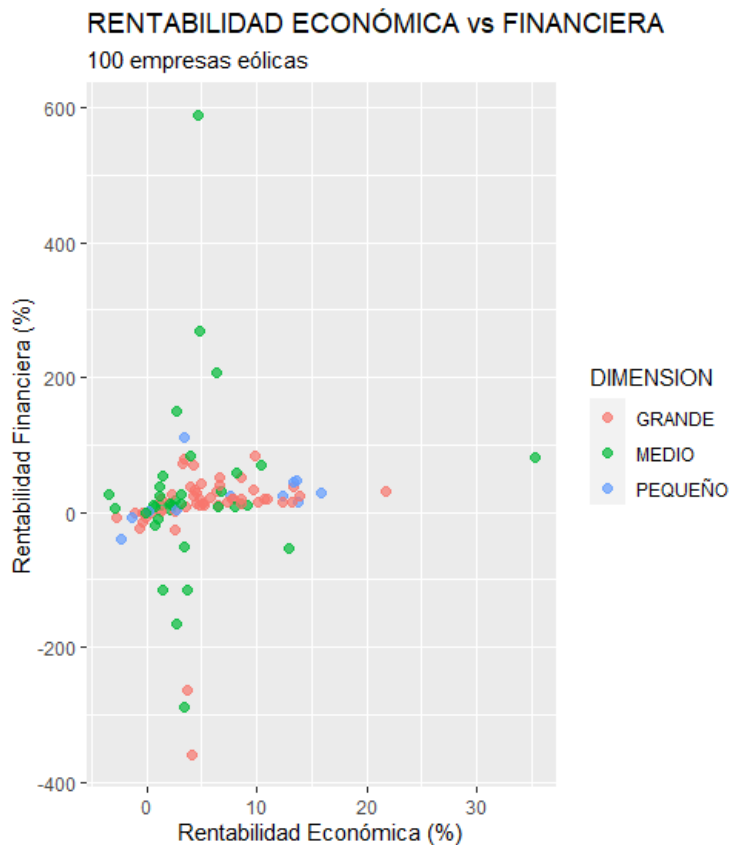
El gráfico obtenido con el código anterior será:



Vamos a refinar el gráfico algo más. En primer lugar, puede ser interesante distinguir entre los grupos de empresas, según su volumen de activos (variable DIMENSION). Para ello, podemos poner el color de los puntos en el "mapeo", en función de la variable DIMENSION:

```
ggplot(data = eolica_100, map = (aes(x = RENEKO, y = RENFIN, col =
DIMENSION))) +
  geom_point(size = 2, alpha = 0.7) +
  ggtitle("RENTABILIDAD ECONÓMICA vs FINANCIERA", subtitle = "100
empresas eólicas") +
  xlab("Rentabilidad Económica (%)") +
  ylab("Rentabilidad Financiera (%)")
```

El resultado, ahora, será:



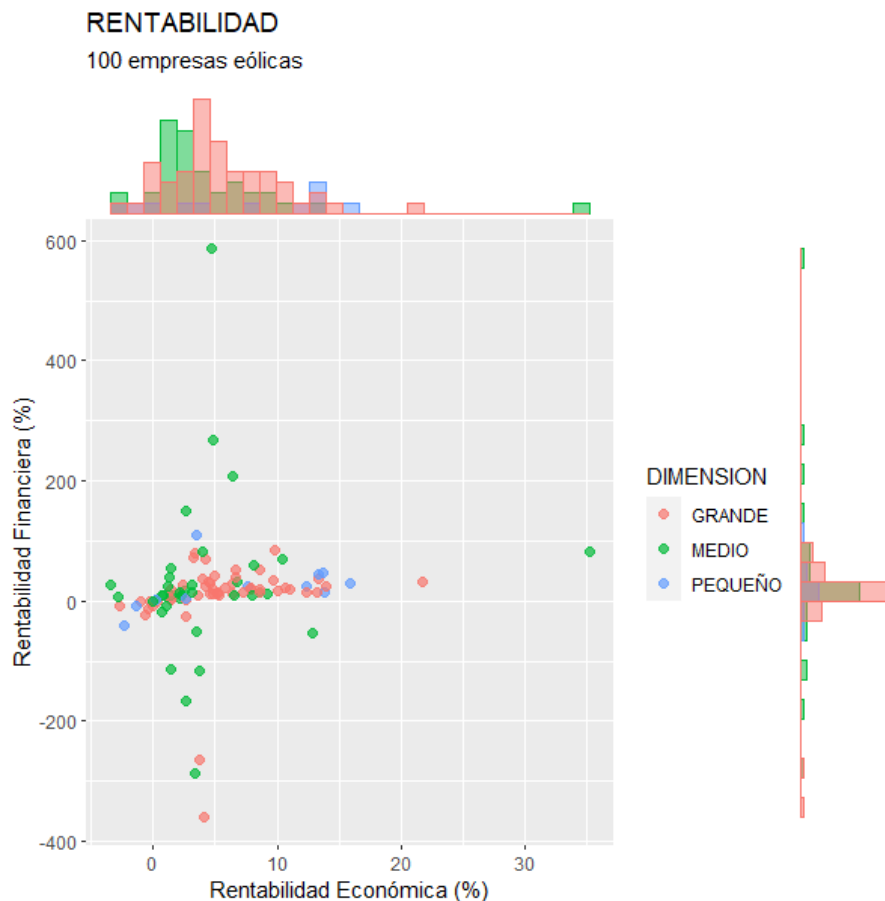
En los dosgráficos anteriores pueden observarse puntos (casos) candidatos a ser *outliers* para cada una de las dos variables analizadas. En el caso de RENEKO, ya se pudo advertir esta circunstancia al construir los *boxplots*.

Podría ser interesante complementar el gráfico con información sobre las dos variables por separado, es decir, con información sobre las distribuciones marginales. Existe un paquete complementario a `ggplot2`, llamado `ggExtra`, que puede ayudar fácilmente a este cometido. Para ello, hemos de activar dicho paquete con `library()` (si no ha sido previamente instalado, habrá que hacerlo con anterioridad). El segundo paso consistirá en **asignar** nuestro *scatterplot*, diseñado con la función `ggplot()`, a un objeto con el nombre que queramos, por ejemplo, "scatter\_plus". Luego,

ese objeto que contiene nuestro gráfico entrará en la función de `ggExtra`, `ggMarginal()`, como se muestra en el siguiente código:

```
library("ggExtra")
scatter_plus <- ggplot(data = eolica_100, map = (aes(x = RENEKO, y =
RENFIN, col = DIMENSION))) +
  geom_point(size = 2, alpha = 0.7) +
  ggtitle("RENTABILIDAD", subtitle = "100 empresas eólicas") +
  xlab("Rentabilidad Económica (%)") +
  ylab("Rentabilidad Financiera (%)")
ggMarginal(scatter_plus, type = "histogram", groupColour = T,
groupFill = T,
position = "identity", alpha = 0.5)
```

Con el código anterior, apreciamos cómo se añaden los histogramas de cada variable, RENEKO y RENFIN, en los márgenes del gráfico.



Adicionalmente, el diámetro de los puntos de los diversos casos podría contener también información, haciéndolo proporcional a una tercera variable. Por ejemplo, podría ser proporcional al nivel de solvencia (variable SOLVENCIA). Para ello, se podría ejecutar el código:

```
scatter_plus <- ggplot(data = eolica_100, map = (aes(x = RENEKO, y =
RENFIN, col = DIMENSION, size = SOLVENCIA))) +
```

```

geom_point(alpha = 0.7) +
ggtitle("RENTABILIDAD", subtitle = "100 empresas eólicas") +
xlab("Rentabilidad Económica (%)") +
ylab("Rentabilidad Financiera (%)")
ggMarginal(scatter_plus, type = "histogram", groupColour = T,
groupFill = T,
position = "identity", alpha = 0.5)

```

En el código anterior, puede comprobarse que la característica “size =” sube del bloque de *geom* al “mapeo” (incluido en el *aes*), debido a que el diámetro de cada punto ya no va a ser un parámetro fijo, sino que va a depender de la magnitud de la variable SOLVENCIA.

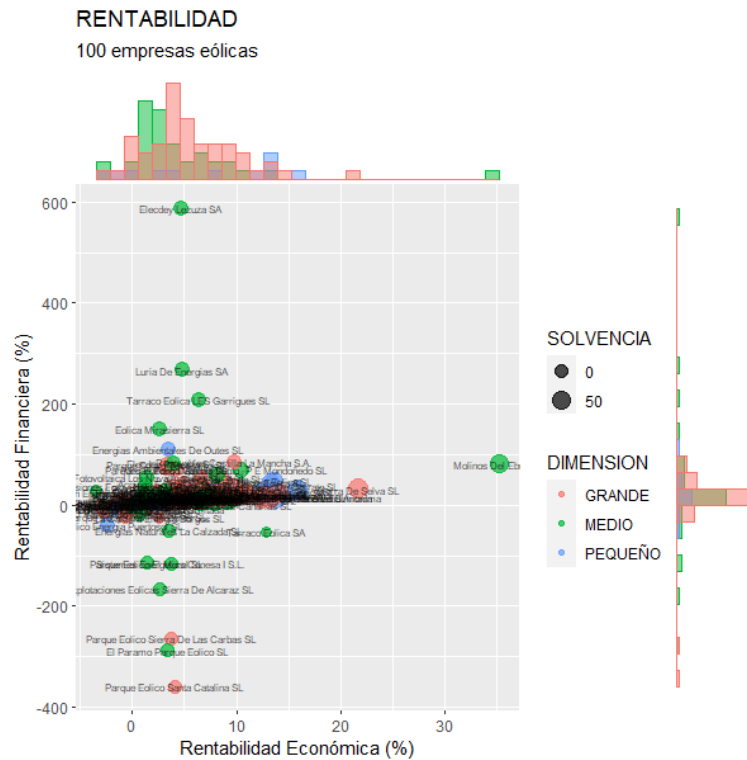
Finalmente, podría ser útil, en algunos gráficos, añadir una etiqueta (*label*) a cada punto, para **identificar el caso concreto** al que representa. Si bien en esta práctica, el elevado número de casos y el extenso nombre de las empresas hacen poco claro el uso de estas etiquetas, vamos a añadirlas por motivos pedagógicos. Para ello, se añadirá un bloque *geom* llamado *geom\_text()*, con una información “label =” que se hace depender de valores que cambian (en concreto, el nombre de los casos, es decir, de las filas del *data frame*), por lo que tendrá que integrarse en una característica *aes*:

```

scatter_plus <- ggplot(data = eolica_100, map = (aes(x = RENEKO, y =
RENFIN, col = DIMENSION, size = SOLVENCIA))) +
geom_point(alpha = 0.7) +
geom_text(aes(label=row.names(eolica_100)), size=2, color="black",
alpha = 0.7) +
ggtitle("RENTABILIDAD", subtitle = "100 empresas eólicas") +
xlab("Rentabilidad Económica (%)") +
ylab("Rentabilidad Financiera (%)")
ggMarginal(scatter_plus, type = "histogram", groupColour = T,
groupFill = T,
position = "identity", alpha = 0.5)

```

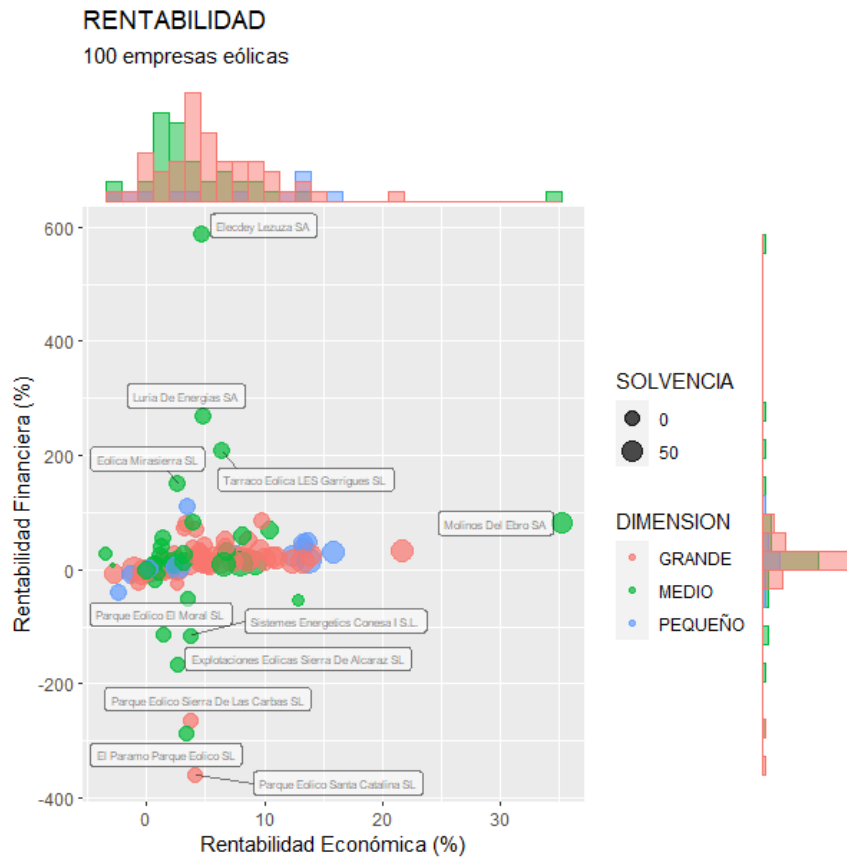
El resultado de todo lo anterior será el siguiente gráfico:



Las etiquetas de los casos pueden refinarse algo más mediante el `geom_label_repel()`, disponible al cargar el paquete `ggrepel()`:

```
library(ggrepel)
scatter_plus <- ggplot(data = eolica_100, map = (aes(x = RENECO, y =
RENFIN, col = DIMENSION, size = SOLVENCIA))) +
  geom_point(alpha = 0.7) +
  geom_label_repel(aes(label = row.names(eolica_100)), size = 2, color
= "black", alpha = 0.5) +
  ggtitle("RENTABILIDAD", subtitle = "100 empresas eólicas") +
  xlab("Rentabilidad Económica (%)") +
  ylab("Rentabilidad Financiera (%)")
ggMarginal(scatter_plus, type = "histogram", groupColour = T,
groupFill = T,
position = "identity", alpha = 0.5)
```

El gráfico construido finalmente será:



La ventaja en este gráfico es que se omiten las etiquetas superpuestas.

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

Updated: 29/09/2022