



# Fastbook 04

## Visual Analytics

Scatter plots



## 04. Scatter plots

En este fastbook vamos a profundizar en uno de los gráficos más utilizados en el mundo del data science: el **scatter plot**. Para ello, comenzaremos analizando el uso y la lógica que hay detrás de este tipo de visualizaciones, desgranando completamente todos los elementos que lo componen. Una vez la teoría esté clara, pasaremos a dibujar nuestros propios diagramas de dispersión con las principales librerías de R.

Es importante que hayas entendido bien la lógica detrás de las librerías que utilizaremos y que vimos en el fastbook anterior para que todo el código que veamos aquí sea ‘digerible’. Además, como venimos hablando, intenta ‘picar’ el código siempre, así es como realmente aprenderás.

*Autor: Daniel Pegalajar Duque*

[Scatter plots o diagrama de dispersión: descripción](#)

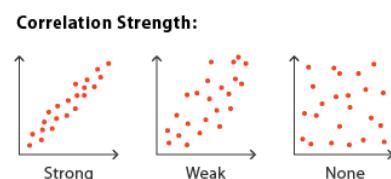
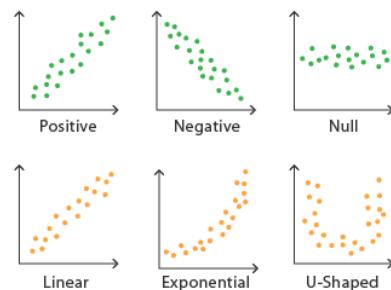
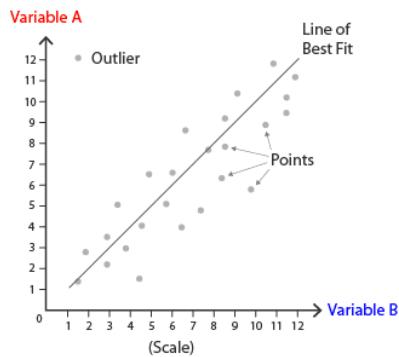
[Scatter plots en R](#)

[Conclusiones](#)

# Scatter plots o diagrama de dispersión: descripción

X Edix Educación

Los **scatter plots** (conocidos también como **diagrama de puntos** o **de dispersión**) son un tipo de visualización que utiliza coordenadas cartesianas para mostrar la relación entre dos variables. Debido a dicho sistema de coordenadas, también reciben el nombre de **gráfico X-Y** o **relacional**.



Fuente: [DataVizCatalogue](#).

Posiblemente, este sea **el gráfico más utilizado en el mundo de los datos para analizar y visualizar la relación entre dos variables**. Es la primera elección que debes tener en mente siempre que quieras responder dicha pregunta.

La potencia de este tipo de visualizaciones es que se pueden descubrir las relaciones de dos variables de un simple vistazo. El data scientist más curtido ha tenido que aprender esto desde sus inicios, por lo que dedica un tiempo a entender perfectamente esta imagen.

Su funcionamiento es muy simple:

- Cada **eje** representa una de las variables de interés (X e Y, altura y peso, edad y rendimiento físico...).
- Cada **punto** u observación de nuestro conjunto de datos es una coordenada en este lienzo, gracias a los valores en las variables seleccionadas.
- Una vez pintados los puntos, se puede ajustar la mejor línea que represente dicha concentración o amalgama de puntos. Este concepto os debería sonar ya: hablamos de la **recta de regresión** (en este caso de Y sobre X).
- Finalmente, este tipo de visualizaciones permite localizar rápidamente la presencia de observaciones atípicas, también conocidas como **outliers**. Este tipo de observación puede ser debida en ocasiones a errores en el grabado de la información o, en el caso de ser correcto, observaciones particulares que deben ser observadas y analizadas con lupa.

Para entender el **sentido** de la posible relación entre las variables basta con mirar la dirección a la que se dirigen los puntos (de color **verde** en la imagen anterior):

- Si su trayectoria va desde el centro del eje de coordenadas hacia la esquina superior derecha, **ambas variables tendrán una relación (correlación) positiva**.
- Si van desde la esquina superior izquierda a la esquina inferior derecha, **la relación será negativa**.
- Si no detectas ninguno de estos patrones y solo se observa una masa de puntos paralela a alguno de los ejes, **dichas variables no están relacionadas linealmente**.

Por otro lado, estaría el **tipo** de relación existente entre las variables. Para entender esta, hay que observar la forma de los puntos (color **amarillo** en la imagen anterior):

- Habitualmente, encontrarás el **patrón lineal** en tus datos. Este patrón se reconoce porque a lo largo de los ejes los puntos no cambian su comportamiento, es homogéneo. Fíjate en el ejemplo de la imagen: si trazásemos la línea de regresión de dichos puntos, todos en la imagen estarían a una distancia muy similar a lo largo de toda la recta.
- Pero no todo en el mundo es lineal, por lo que podrás encontrar formas muy llamativas, como **comportamientos exponenciales o totalmente no lineales**. Recuerda que, en estos casos, la correlación no es útil, ya que esta métrica solo sirve para relaciones lineales.

Finalmente, según la dispersión de los puntos podemos asegurar visualmente una **intensidad** en la relación entre ambas variables (color **rojo** en la imagen anterior):

- Mientras más se asemejen los puntos a una **línea recta de  $45^{\circ}$** , más intensa es la correlación entre ambas variables. De hecho, la **correlación perfecta** (igual a 1 o -1) visualmente luciría como una línea de puntos perfecta de  $45^{\circ}$ .
- Conforme se **dispersan los puntos**, la **intensidad relacional** se desvanece y nos acercamos a la frontera de **correlación igual a 0**. Para entender cuando no existe relación alguna entre dos variables, hay que imaginar una nube de puntos donde aparentemente no se observa patrón alguno.

El ejemplo más visual que siempre se me ocurre es el siguiente: coge un puñado de lentejas, déjalo caer encima de la mesa (sin formar un estropicio): lo que obtendrás es una masa de puntos sin relación alguna.

Ahora que ya tenemos desmigado este tipo de visualización, comenzemos a profundizar en sus posibilidades utilizando nuestras herramientas.

# Scatter plots en R

X Edix Educación

Comencemos nuestra andadura en este fastbook con R. Este tipo de gráfico ya lo viste en el fastbook anterior y fue ‘tu primer gráfico’. Ahora vamos a profundizar.

```
# Desactivamos la notación científica. ¿A quién le gusta ver en sus gráficos números como
# 1e25?
options(scipen = 999)

# Cargamos las librerías necesarias para pintar
library(ggplot2) # Nuestra biblia a partir de ahora
library(scales) # Nos ayudará a mejorar el aspecto de nuestros gráficos

library(tidyverse) # Necesario si queremos realizar algún tratamiento en los datos

# Establecemos un tema por defecto para nuestros gráficos
# Personalmente soy fanático de theme_bw(), es el tema clásico 'dark-on-light'
# ggplot ofrece un listado de temas completos que puedes aprovechar. Echa un vistazo:
# https://ggplot2.tidyverse.org/reference/ggtheme.html

# Hay gente que realiza sus propios temas, generando auténticas obras de arte, ¿te atreves?
theme_set(theme_bw())

# Cargamos los datos que vamos a utilizar

data("diamonds", package = "ggplot2")
data("txhousing", package = "ggplot2")
```

Lo primero es cargar las librerías obligatorias y configurar algunas opciones por defecto para obtener mejores resultados, como es el caso de la posible notación científica o el tema a utilizar por defecto en ggplot.

Los **datos** que se cargarán en memoria para este ejercicio ya vienen implementados en la propia librería **ggplot2**. Vamos a dar algún detalle adicional:

- **Diamonds:** este conjunto de datos viene incorporado con la librería ggplot2 y para entender mejor sus variables, puedes utilizar la orden `help(diamonds)`. Básicamente, contiene información sobre el precio y otros atributos de alrededor de 54.000 diamantes. Seguro que te suena ya del fastbook anterior.

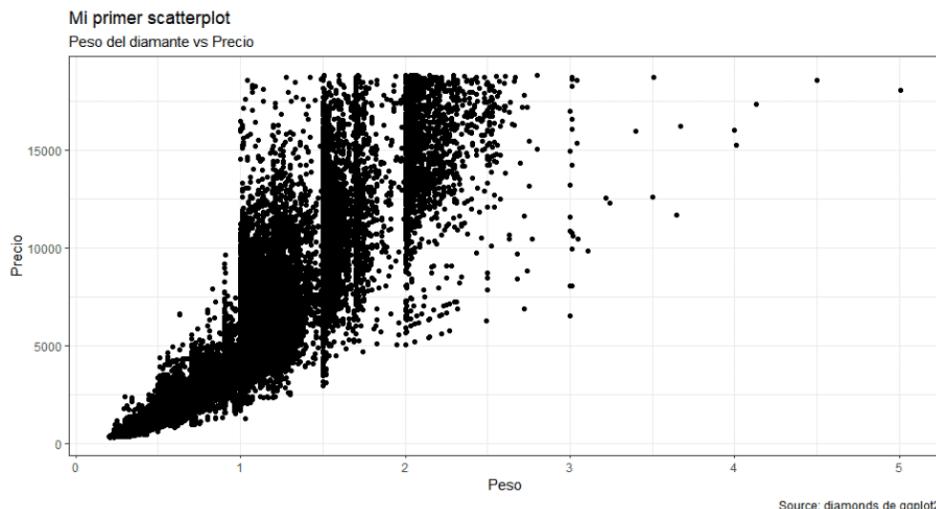
- **Txhousing:** este dataset contiene 8602 observaciones y 9 variables que hacen referencia a información sobre el mercado inmobiliario del mercado de Texas. Todos estos datos provienen de [TAMU](#).

En ggplot, la orden que nos ayudará a conseguirlo es `geom_point()` que solo requiere como input dos variables numéricas. Además, podemos aprovechar la función `geom_smooth()`, que superpone una recta de regresión sobre esos puntos, ayudándonos a entender la intensidad de dicha relación. ¿A que te va sonando?

Vamos a verlo:

```
## Empezamos con lo simple
gg ← ggplot(diamonds, aes(x = carat, y = price)) +
  geom_point() +
  labs(title = "Mi primer scatterplot",
       subtitle = "Peso del diamante vs Precio",
       x = "Peso",
       y = "Precio",
       caption = "Source: diamonds de ggplot2")
gg
```

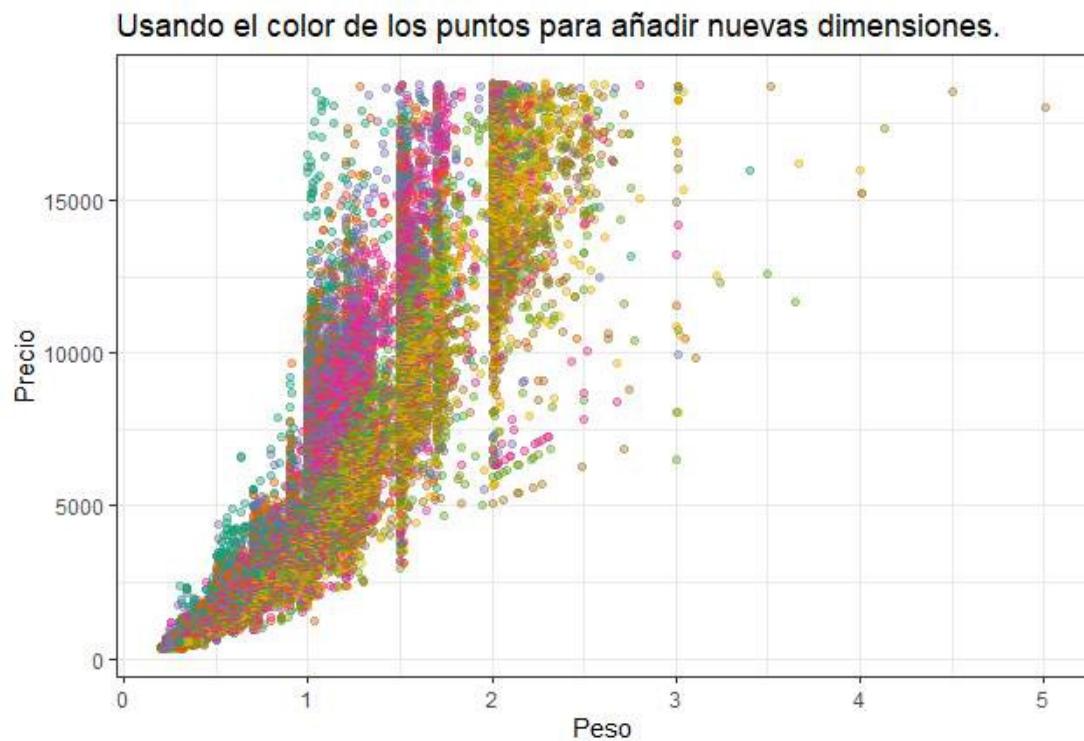
Retomamos nuestro análisis de los diamantes. Sospechamos que existe una relación entre el peso del diamante (`carat`) y su precio en el mercado. Utilizamos un scatter plot para demostrar nuestra hipótesis. Por último, observa la función `labs()`: nos permite añadir y enriquecer la **información que mostrará el gráfico**. El resultado es el siguiente:



Este gráfico no debería sorprenderte en exceso, en el anterior fastbook ya deberías haber conseguido uno muy similar.

Es hora de comenzar a enriquecer la visualización y extraer todo el potencial de ggplot haciendo uso de los **aesthetics**. Comencemos añadiendo una tercera dimensión en base al color del diamante:

```
## Añadimos una nueva dimensión a los datos
gg <- ggplot(diamonds, aes(x = carat, y = price, col = color)) +
  geom_point(alpha = 0.4) +
  scale_color_brewer(palette = "Dark2") +
  labs(x = "Peso", y = "Precio",
       title = "Usando el color de los puntos para añadir nuevas dimensiones.")
gg
```



En este paso hemos introducido varias órdenes nuevas, vamos a repasarlas:

- Como se aprecia, utilizando el **aesthetic** col hemos requerido una nueva dimensión, el color del diamante. Al indicar el nombre de la variable en la fuente de datos, la librería se encarga de mapear los 54.000 diamantes y pintarlos en función al color que tengan asignado.

- Hemos incluido el argumento **alpha** en *geom\_point()*. Esta orden se utiliza para añadir transparencia a los puntos representados.  
Va desde 0 a 1: mientras más próximo a 0, más transparencia obtendremos. Suele ser útil cuando se tiene un gran número de elementos en un gráfico (recuerda que en este caso estamos pintando 54.000 puntos...).
- Por último, hemos definido la **paleta de colores a utilizar**. No pasa nada si no especificas esta última línea ya que ggplot entenderá que dejas en sus manos la elección de colores y usará la paleta por defecto. Si quieres más detalle sobre las paletas disponibles utiliza el *help()* con esa función.

Ahora analicemos el gráfico resultante. Tras añadir la capa de color es interesante buscar si la inclusión de esta nueva dimensión añade aprendizajes a lo anteriormente visto. Se puede observar que los colores representan los diferentes grados de color de cada diamante y que, como indica la leyenda, van desde la D (lo mejor), hasta la J (peor). Un detalle sobre el color de los diamantes:



Fuente: [GIA](#)

---

En nuestro conjunto de datos tenemos hasta la letra J. En la fuente que te dejo debajo de la imagen podrás profundizar más, pero básicamente, conforme bajamos en la escala, el diamante adquiere tonos amarillentos, lo que incide en su precio.

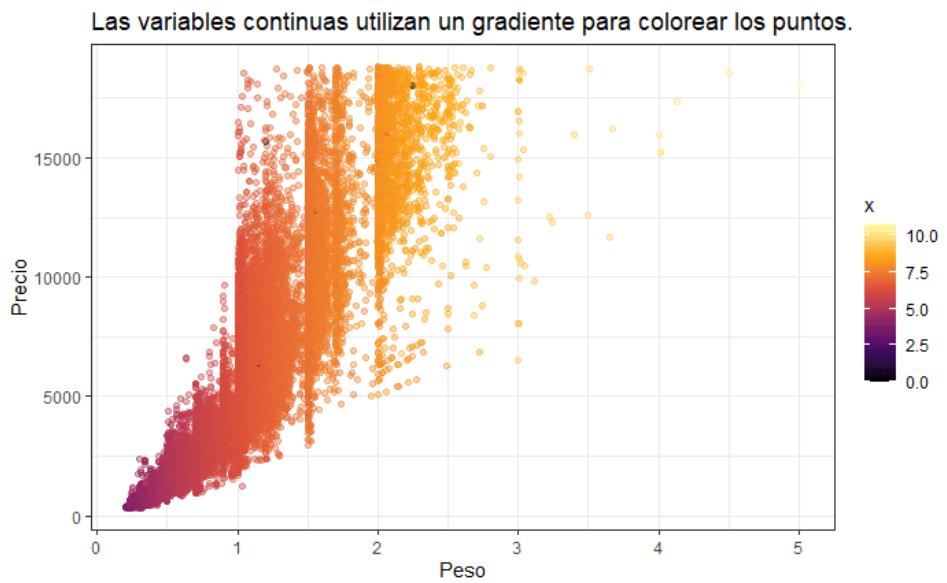
Se aprecia cómo las letras más cercanas al grado D poseen un mayor precio que la franja inferior donde se acumulan la mayoría de los diamantes H, I y J. De hecho, es fácilmente apreciable una buena separación en los grados de color.

## ¿Lo ves? ¿Identificas algún punto con buen color, pero alejado de su grupo?

Si te fijas, la variable color es una variable categórica (sus valores representan dimensiones exhaustivas y excluyentes, como el color de los ojos). Eso hace que la escala de color seleccionada para representar el gráfico sea discreta. Si en lugar de una variable categórica seleccionamos una numérica, se usará una escala de color continua o gradiente para representar dicha escala.

Por ejemplo, si utilizamos la variable X como color:

```
## Añadimos una nueva dimensión a los datos
gg ← ggplot(diamonds, aes(x = carat, y = price, col = x)) +
  geom_point(alpha = 0.4) +
  # scale_color_brewer(palette = "Dark2") + | 
  scale_colour_viridis_c(option = "B") +
  labs(x = "Peso", y = "Precio",
       title = "Usando el color de los puntos para añadir nuevas dimensiones.")
gg,
```



Podemos utilizar otra variable numérica para detectar patrones en los datos. Usando esta variable observamos un **comportamiento creciente** conforme el diamante aumenta el peso/precio. También detectamos ciertos outliers en la zona superior que están fuera de lugar, ¿los ves?

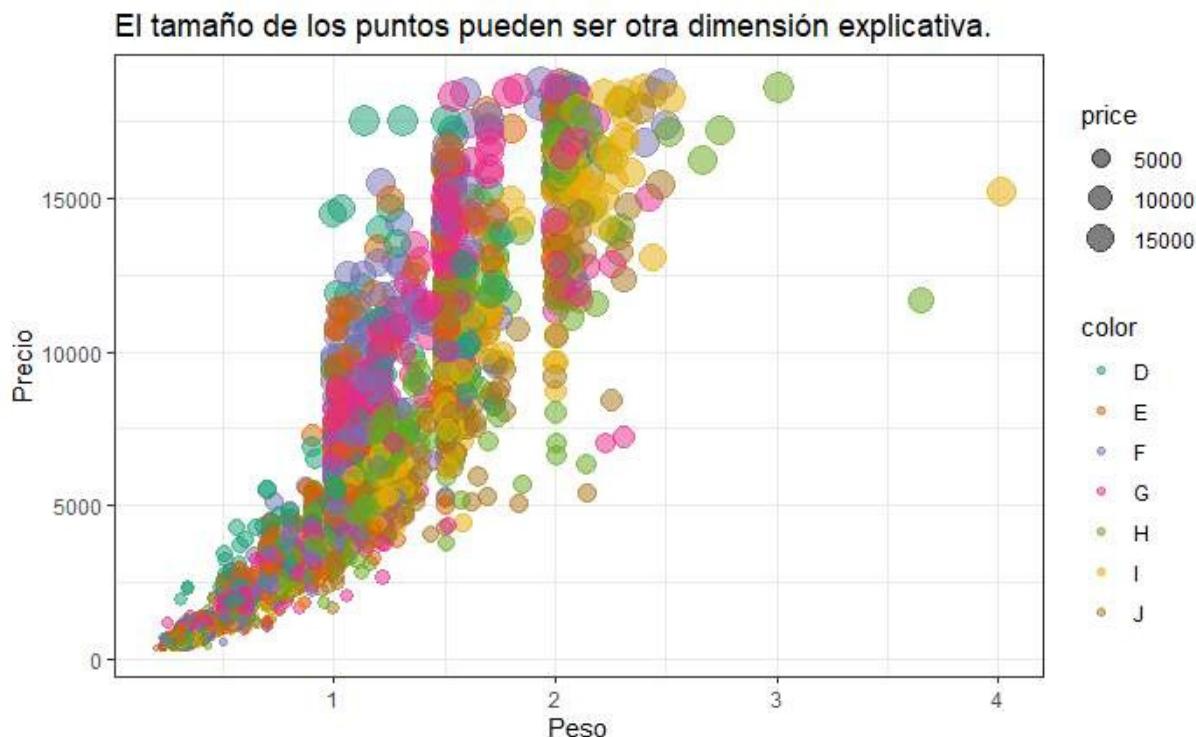
---

Hemos usado la función `scale_colour_viridis_c()` para pintar con la paleta viridis. Echa un vistazo en la ‘help’ para obtener más detalles.

Por último, puedes explotar otras dimensiones que te permitan añadir nuevas capas de información a tu visualización. Por ejemplo, respecto al punto anterior, todavía puedes aprovechar el tamaño del punto y su forma para vincularlos a otros atributos. Veamos un ejemplo más utilizando el tamaño del punto:

```
## Añadimos una nueva dimensión a los datos
gg <- ggplot(diamonds %>% sample_n(5000), aes(x = carat, y = price, col = color, size = price)) +
  geom_point(alpha = 0.5) +
  scale_color_brewer(palette = "Dark2") +
  # scale_colour_viridis_c(option = "B") +
  labs(x = "Peso", y = "Precio",
       title = "El tamaño de los puntos pueden ser otra dimensión explicativa.")
```

gg



Atiende en este último paso al uso de la función `sample_n()` para elegir una muestra del dataset original para representarlos. Cuando se trabaja con conjuntos de datos muy grandes, puede ser un alivio para el ordenador y la representatividad no se verá muy afectada. Siempre hay que usarlo con precaución. Otra función similar es `sample_frac()` donde se usa un porcentaje de datos que quieras retener.

---

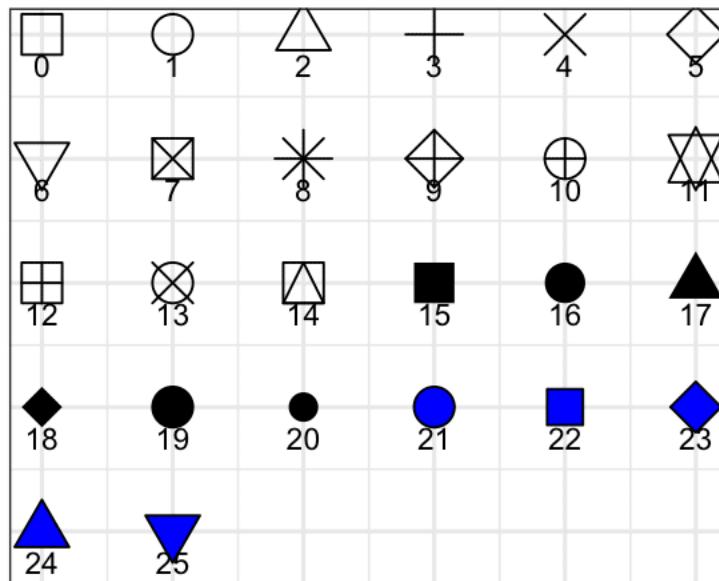
## ¡Realiza tus propias pruebas y experimenta con otras variables!

Como nota adicional para la dimensión de forma del punto, aquí te dejo una forma de visualizarlos:

```
ggsignif::show_point_shapes() #
```

Es necesario instalar el paquete ggsignif.

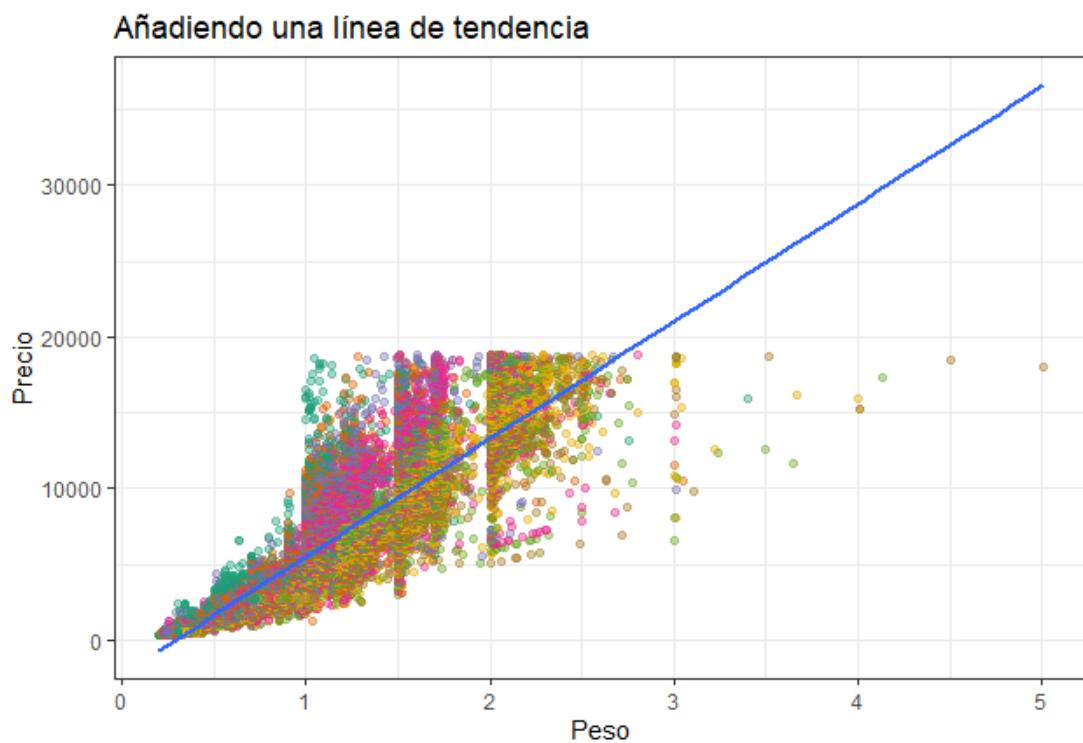
### Point shapes available in R



Si queremos enriquecer aún más un scatter plot, podemos añadir a la visualización **curvas de tendencia** que nos ayuden a explicar o detectar los patrones en los datos. Esto lo conseguimos en R mediante la función `geom_smooth()` que, mediante diferentes métodos de regresión, genera la mejor recta/curva que representa las observaciones.

Vamos a verlo mediante un ejemplo:

```
## Añadimos una nueva dimensión a los datos
gg ← ggplot(diamonds, aes(x = carat, y = price)) +
  geom_point(aes(col = color), alpha = 0.4) +
  geom_smooth(se = F, method = "lm") +
  scale_color_brewer(palette = "Dark2") +
  labs(x = "Peso", y = "Precio", col = "Color",
       title = "Añadiendo una línea de tendencia")
gg
```



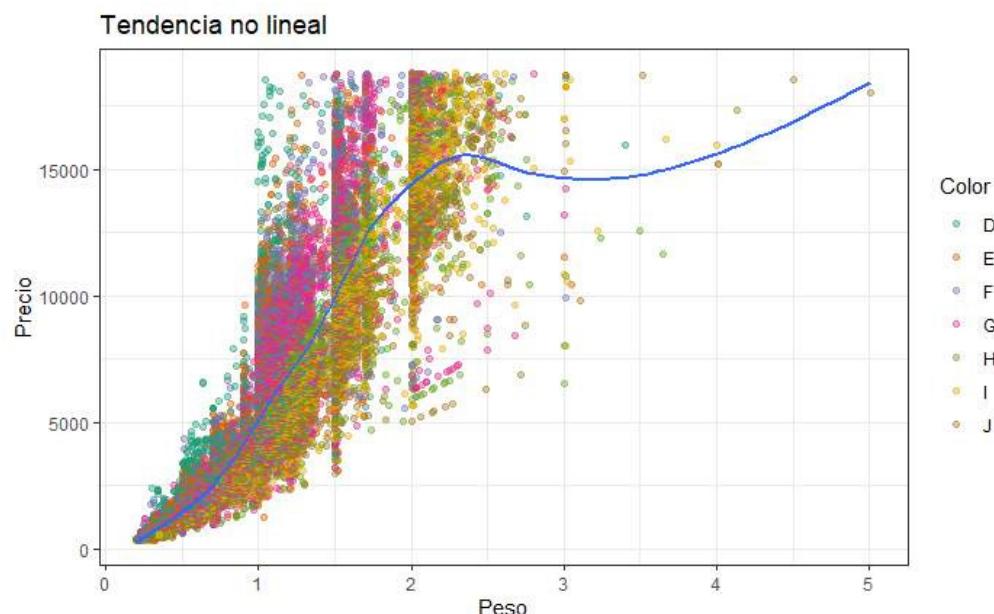
Hemos utilizado los argumentos `se = False`, que elimina las bandas de confianza sobre la línea de tendencia y `method = "lm"`, que especifica el uso de la regresión lineal para generar dicha tendencia. Puedes probar diferentes métodos consultando la ayuda de `geom_smooth()`.

Algunos apuntes a tener en cuenta:

- Si te fijas bien, hemos cambiado el `aes()` de color y lo hemos introducido dentro de `geom_point()`, ¿por qué? Los argumentos estéticos son hereditarios en `ggplot` por lo que, lo que defines en la primera orden de `ggplot()` se arrastra al resto de funciones `geom_`. Esto quiere decir que si mantenemos la orden de color en la primera orden, la nueva función de `geom_smooth()` se realizará para cada una de esas dimensiones (es decir, una línea para cada dimensión del color del diamante).
- Hemos representado una recta de regresión mediante el método `lm`. Dicha inclinación tan pronunciada nos demuestra la relación que sospechábamos: a mayor peso, mayor precio tendrá el diamante. Pero no parece muy realista a la vista de los datos, donde se aprecia que no siempre va a ser lineal y que llega un precio en el que se frena esa relación. **Podemos probar a utilizar otras funciones de regresión** para representar esta relación.
- Dividir esta foto para cada color nos permitiría entender en qué colores la relación es más fuerte y si existen algunos que carezcan de dicha relación.

```
## Cambiamos el tipo de recta
gg <- ggplot(diamonds, aes(x = carat, y = price)) +
  geom_point(aes(col = color), alpha = 0.4) +
  geom_smooth(se = F, method = "gam") +
  scale_color_brewer(palette = "Dark2") +
  labs(x = "Peso", y = "Precio", col = "Color",
       title = "Tendencia no lineal")

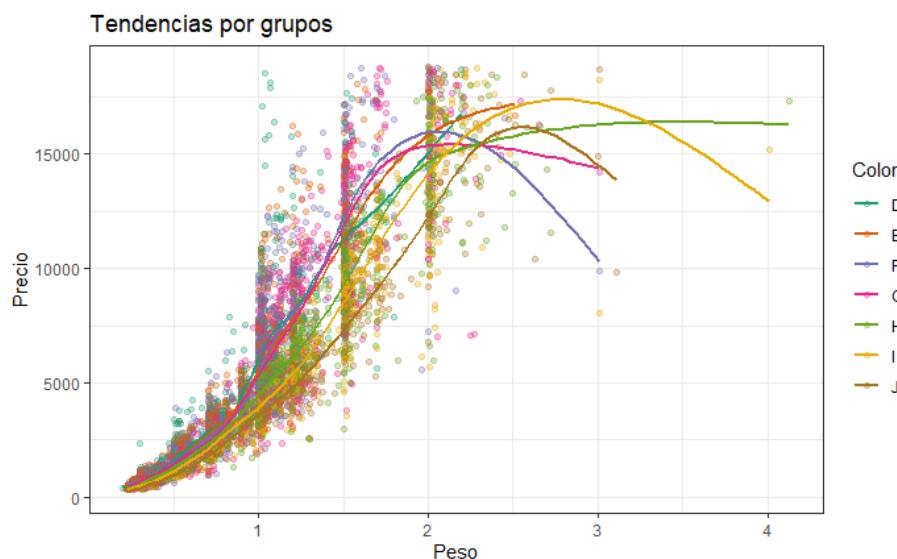
gg
```



En este caso hemos cambiado la tendencia a una no lineal para mejorar el comportamiento. Aun así, la presencia de outliers a partir de los 3 gramos, empeora la confianza de dicha curva y deberíamos tener cuidado con las posibles interpretaciones. Finalmente, ilustramos la generación de curvas de tendencia por grupos:

```
## Buscamos cada color
gg <- ggplot(diamonds %>% sample_n(10000), aes(x = carat, y = price, col = color)) +
  geom_point(alpha = 0.3) +
  geom_smooth(se = F, method = "gam") +
  scale_color_brewer(palette = "Dark2") +
  labs(x = "Peso", y = "Precio", col = "Color",
       title = "Tendencias por grupos")
```

```
gg
```



Al subir el argumento `col` a la función principal de `ggplot()` se hace hereditario para todas las funciones venideras. Con esto conseguimos que, tras llamar a `geom_smooth()`, la generación de tendencias se haga por grupo específico. Esto nos permite sacar aprendizajes directos sin tener que pelear por localizar visualmente a todos los grupos. Podemos ver que los diamantes de letras más cercanas a E poseen mejor precio, mientras que H, I o J están algo por debajo.

---

**Te animo a que practiques lo aprendido y utilices variables diferentes para generar nuevas visualizaciones y posibles aprendizajes.**

# Conclusiones

X Edix Educación

---

En el cuarto fastbook hemos iniciado nuestro recorrido profundizando en uno de los gráficos más utilizados en el mundo del data science, el scatter plot. A través de las tres opciones, hemos generado código para explotar lo que esta visualización puede ofrecer. Como en el anterior fastbook, te recomiendo encarecidamente que pruebes los resultados en cada opción y no te quedes solo con la lectura de este documento. Esta ciencia se aprende con práctica, así que, adelante, intenta picar el código para afianzar lo aprendido.

---

**¿Te atreves a utilizar esta visualización con algún conjunto de datos propio? Prueba y extrae conclusiones.**

A lo largo del tema hemos aprendido el uso de **dimensiones adicionales** que nos ayuden a reforzar nuestras hipótesis e incluso generar nuevos aprendizajes.



Consulta más información en la página principal de [ggplot2](#).

¡Enhорabuena! Fastbook superado

edix

Creamos Digital Workers