Análisis Exploratorio de Datos (EDA)

(basado en el Capítulo 7 del libro "R para Ciencia de Datos' ')

Sebastián Castro

Seminario: R para Ciencia de Datos - Facultad de Ciencias, Universidad de la República

21 y 28/4/2021

Resumen

En este capítulo se muestra cómo usar la visualización y la transformación para explorar los datos de manera sistemática.

En particular, se hará énfasis en la visualización de distribuciones de variables categóricas y continuas, y en la visualización de relaciones entre dos variables (distinguiendo según el tipo de cada una de ellas).

En todos los casos se intentará utilizar estas herramientas para detectar patrones o aspectos llamativos e interesantes de los datos.

Introducción

El Análisis Exploratorio de Datos (EDA) es un ciclo iterativo en el que:

1. se generan preguntas acerca de los datos;
2. se buscan respuestas visualizando, transformando y modelando los datos;
3. se usa lo anterior para refinar y/o generar nuevas preguntas.

El EDA no es un proceso formal regido por un conjunto estricto de reglas, sino más bien donde uno debería ser "libre" de investigar todas las ideas que se le ocurran.

Herramientas en R

En este capítulo se utilizará lo aprendido sobre `dplyr` y `ggplot2` en los capítulos anteriores para la manipulación y visualización de los datos.

```
# Comenzamos entonces cargando los paquetes necesarios

library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(datos)
```

Datos de `diamantes`

```
?diamantes
```

Descripción: Un conjunto de datos que contiene los precios (y otras características) de casi 54.000 diamantes.

Formato: Un data.frame con 53.940 filas y 10 columnas

Variables: `precio`, `quilate` (peso), `corte` (calidad), `color` (de J peor a D mejor), `claridad`, `profundidad` ($2 * z/(x + y)$), `tabla`, `x` (largo en mm), `y` (ancho en mm) y `z` (profundidad en mm).

Preguntas (a los datos)

"*Es preferible una respuesta aproximada a la pregunta correcta, que frecuentemente es formulada de manera imprecisa, que una respuesta exacta a la pregunta incorrecta, que siempre puede ser formulada de manera precisa.*" — John Tukey

El objetivo durante el EDA es desarrollar un entendimiento de los datos. La manera más fácil de lograrlo es usar preguntas como herramientas para guiar la investigación.

Si bien no hay reglas específicas, hay dos tipos de preguntas que siempre serán útiles para hacer descubrimientos sobre los datos:

1. ¿qué tipo de variación existe dentro de cada una de las variables?;
2. ¿qué tipo de covariación ocurre entre diferentes variables?

Algunos términos básicos

Para facilitar la discusión, definamos algunos términos:

(i) variable: una cantidad, cualidad o característica que se puede medir;

(ii) valor: el estado de la variable en el momento en que fue medida (puede cambiar de una medición a otra);

(iii) observación: un conjunto de mediciones realizadas en condiciones similares (usualmente son realizadas al mismo tiempo y sobre el mismo objeto).

Variación (distribución de una variable)

La variación es la tendencia de los valores de una variable a cambiar de una medición a otra.

La mejor manera de entender dicho patrón es visualizando la distribución de los valores de la variable.

Cómo visualizar la distribución de una variable dependerá del tipo de variable que sea:

(i) categórica: si únicamente puede tomar un valor correspondiente a un conjunto finito de valores;

(ii) continua: si puede adoptar un conjunto infinito de valores ordenados.

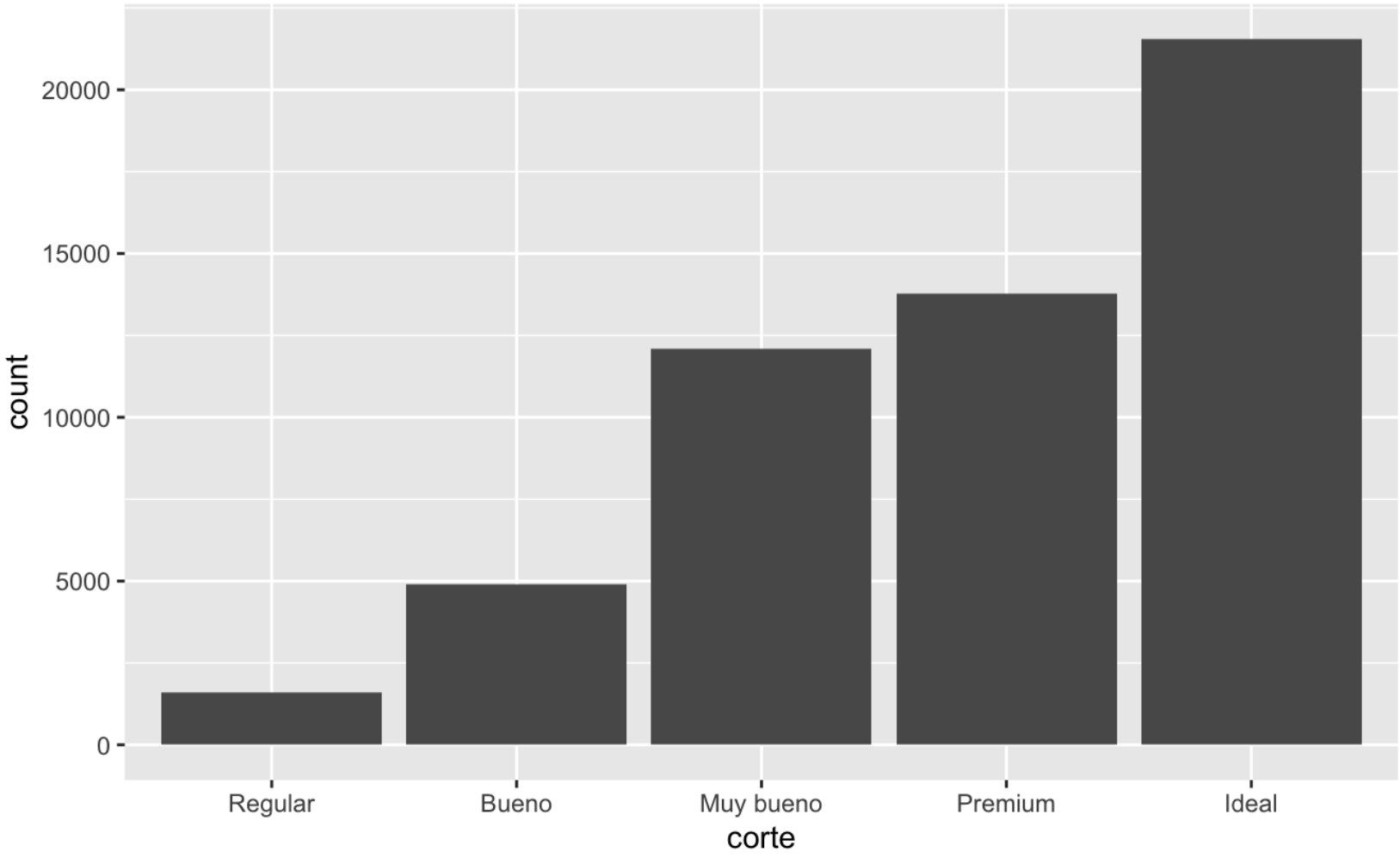
Visualizando distribuciones: ejemplo variable categórica

En R las variables categóricas usualmente son guardadas como factores o vectores de caracteres.

Para examinar la distribución de una variable categórica, se suele utilizar un gráfico de barras de frecuencias:

```
ggplot(data = diamantes) +
  geom_bar(mapping = aes(x = corte))
```

Ejemplo de gráfico de barras de frecuencias



Ejemplo de gráfico de barras de frecuencias

La altura de las barras muestra cuántas observaciones corresponden a cada valor de la variable.

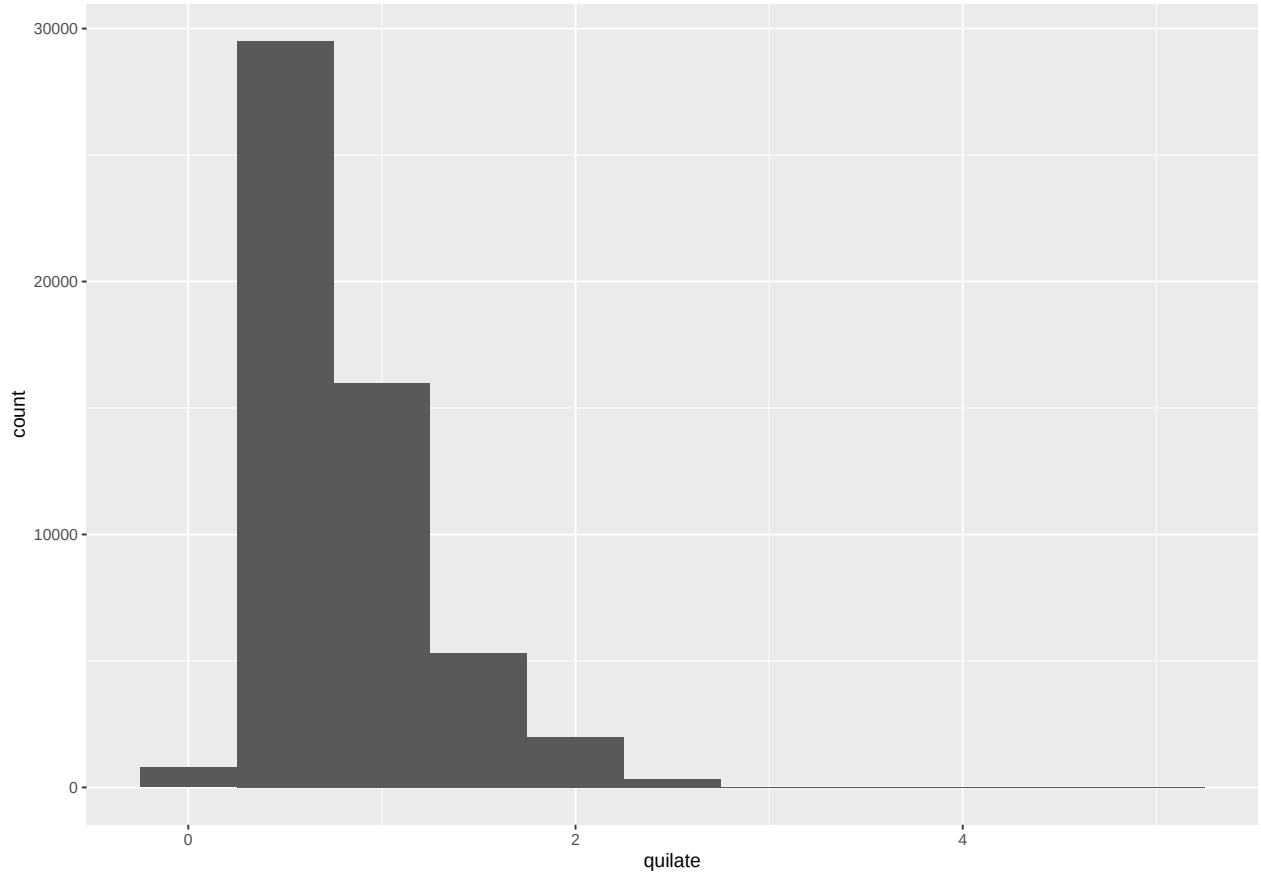
```
diamantes %>%
  count(corte)
```

```
## # A tibble: 5 x 2
##   corte          n
##   <ord>      <int>
## 1 Regular     1610
## 2 Bueno       4906
## 3 Muy bueno  12082
## 4 Premium    13791
## 5 Ideal      21551
```

Visualizando distribuciones: ejemplo variable continua

Para examinar la distribución de una variable continua, usamos un histograma:

```
ggplot(data = diamantes) +
  geom_histogram(mapping = aes(x = quilate), binwidth = 0.5)
```



Ejemplo histograma

La altura de las barras se corresponde con las frecuencias de la variable agrupada en intervalos.

```
diamantes %>%
  count(cut_width(quilate, 0.5))
```

```
## # A tibble: 11 x 2
##    `cut_width(quilate, 0.5)`     n
##    <fct>                     <int>
##  1 [-0.25,0.25]                785
##  2 (0.25,0.75]               29498
##  3 (0.75,1.25]               15977
##  4 (1.25,1.75]                5313
##  5 (1.75,2.25]                2002
##  6 (2.25,2.75]                 322
##  7 (2.75,3.25]                  32
##  8 (3.25,3.75]                   5
##  9 (3.75,4.25]                   4
## 10 (4.25,4.75]                   1
## 11 (4.75,5.25]                   1
```

El histograma

Segmenta el eje horizontal en rangos equidistantes y en la altura de la barra muestra el número de observaciones que corresponden a cada intervalo.

Se puede establecer el ancho de los intervalos de un histograma con el argumento `binwidth`.

Conviene explorar distintas medidas para el ancho del intervalo porque cada una de ellas puede revelar diferentes patrones.

¿Qué deberíamos buscar en estos gráficos?, ¿qué tipo de preguntas deberíamos hacernos?

Valores típicos

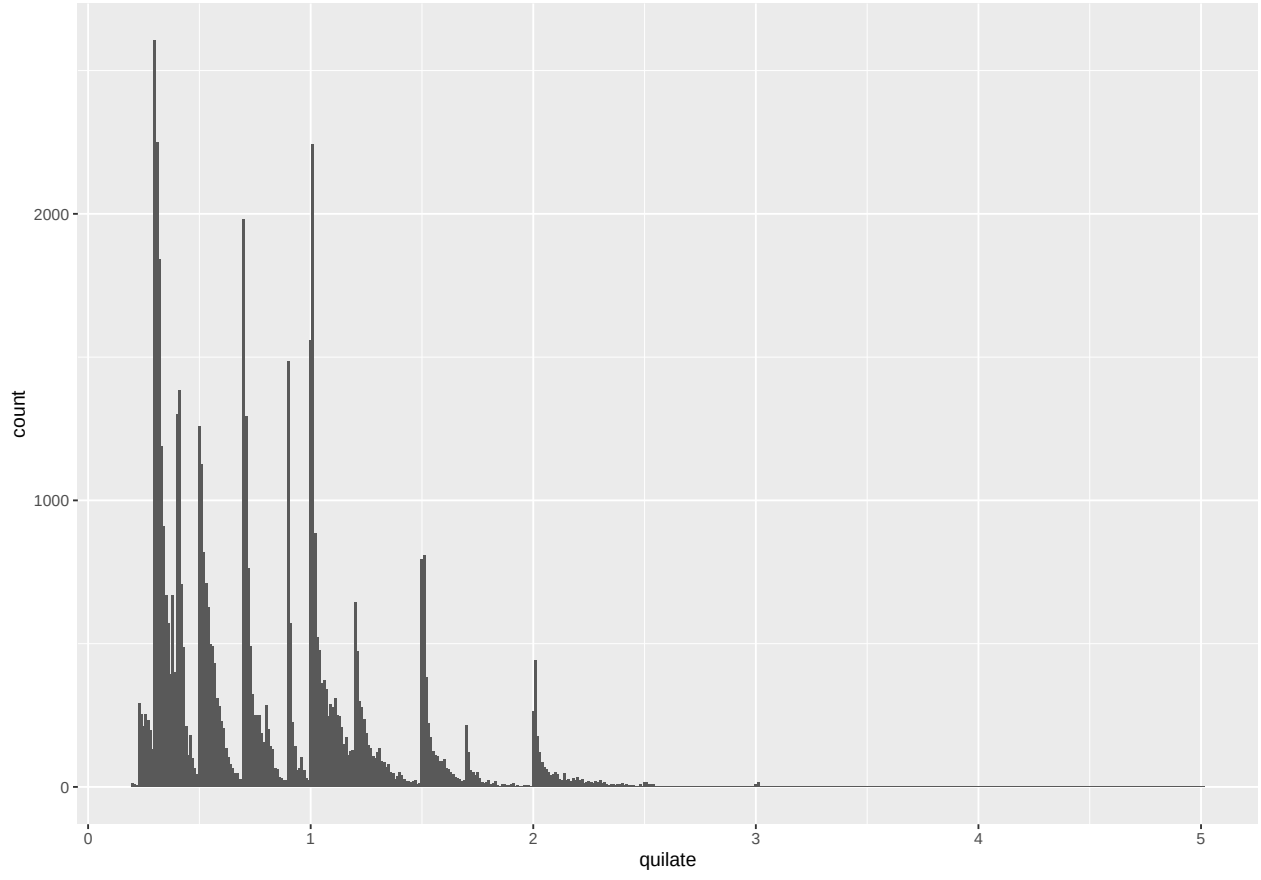
(i) ¿Qué valores son los más comunes?, ¿por qué?;

(ii) ¿Qué valores son infrecuentes?, ¿por qué? ¿Cumple esto con las expectativas?;

(iii) ¿Pueden verse patrones inusuales?, ¿qué podría explicarlos?

Ejemplo

```
ggplot(data = diamantes, mapping = aes(x = quilate)) +
  geom_histogram(binwidth = 0.01)
```



Algunas preguntas que surgen

(i) ¿Por qué hay más diamantes en quilates completos y fracciones comunes de quilates?;

(ii) ¿Por qué hay más diamantes hacia la derecha de cada sección que hacia la izquierda?;

(iii) ¿Por qué hay tan pocos diamantes con más de 3 quilates?

Observar que algunas de estas preguntas o patrones surgieron al cambiar el parámetro `binwidth` en el histograma.

Valores atípicos (*outliers*)

Son valores en los datos que parecen no ajustarse al patrón general (demasiado grandes, demasiado chicos, muy alejados del resto, etc).

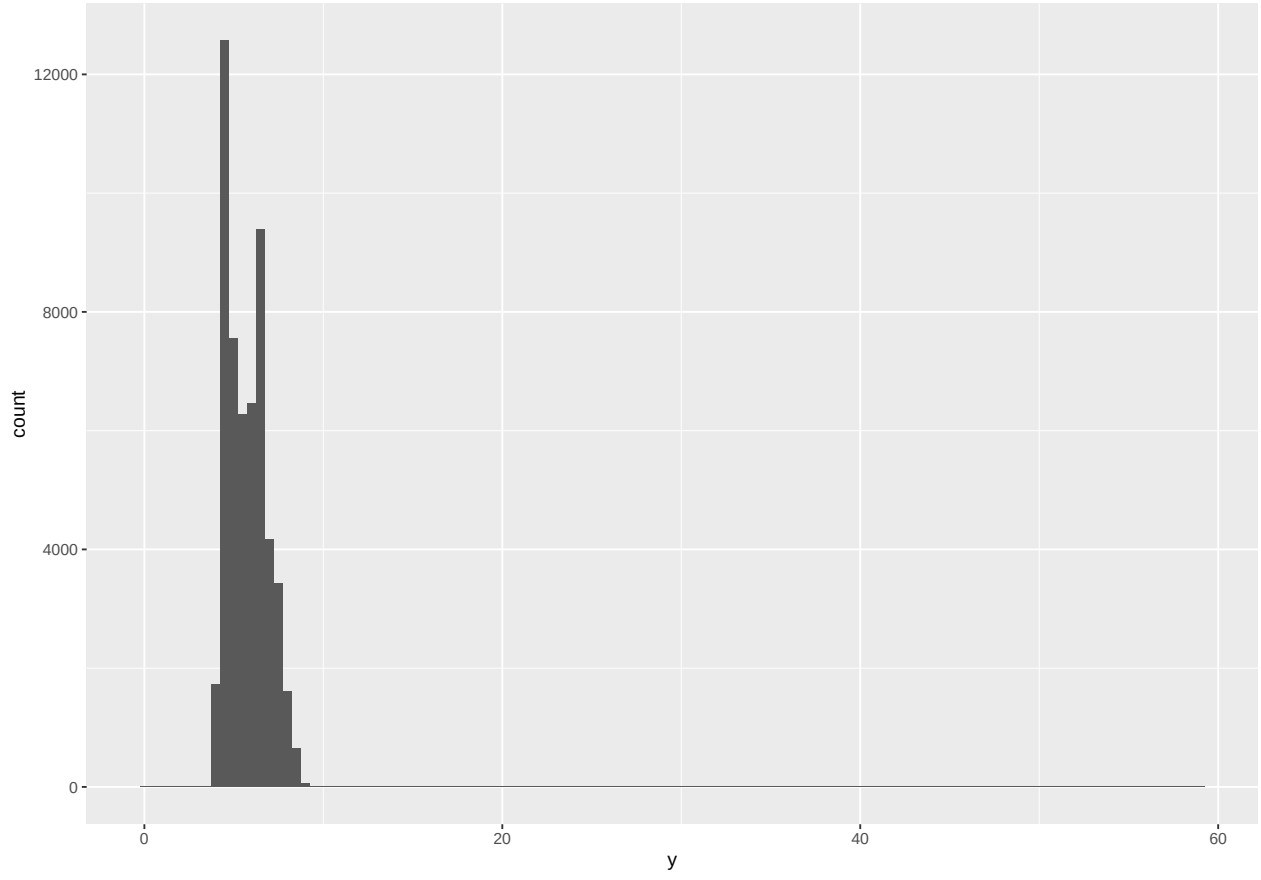
Algunas veces dichos se corresponden con errores cometidos durante el registro de datos.

Otras veces pueden sugerir información relevante.

Cuando se tiene una gran cantidad de datos, es difícil identificar los valores atípicos en un histograma.

Ejemplo: histograma y valores atípicos

```
ggplot(diamantes) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5)
```



Ejemplo: histograma y valores atípicos

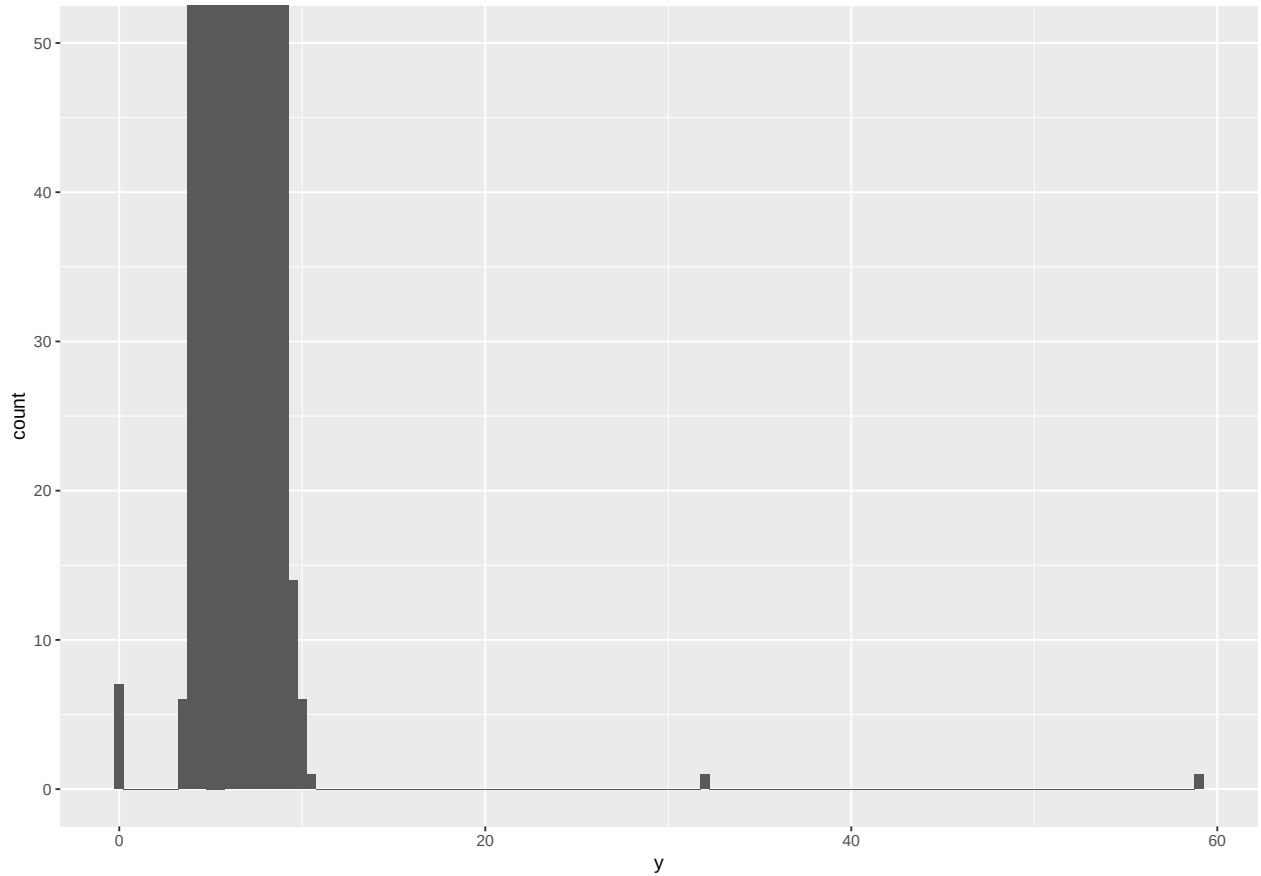
La única evidencia de la existencia de valores atípicos son límites inusualmente anchos en el eje horizontal.

Hay tantas observaciones en los intervalos más comunes (barras más altas) que los intervalos poco frecuentes tienen barras tan cortas que no es posible verlas a simple vista.

Una posible solución consiste en acercar la imagen a los valores más pequeños del eje vertical con `coord_cartesian`.

Ejemplo: histograma y valores atípicos

```
ggplot(diamantes) +
  geom_histogram(mapping = aes(x = y), binwidth = 0.5) +
  coord_cartesian(ylim = c(0, 50))
```



Esto nos permite ver que hay tres valores inusuales: 0, ~30, y ~60, los cuales podemos remover con `filter()`.

Análisis de valores atípicos

```
atipicos <- diamantes %>%
  filter(y < 3 | y > 20) %>%
  select(precio, x, y, z) %>%
  arrange(y)
atipicos
```

```
## # A tibble: 9 x 4
##   precio     x     y     z
##    <int> <dbl> <dbl> <dbl>
## 1   5139  0      0    0
## 2   6381  0      0    0
## 3  12800  0      0    0
## 4  15686  0      0    0
## 5  18034  0      0    0
## 6   2130  0      0    0
## 7   2130  0      0    0
## 8   2075  5.15  31.8  5.12
## 9  12210  8.09  58.9  8.06
```

La variable `y` mide el ancho de estos diamantes en milímetros, así que sabemos que no pueden haber valores de 0 mm. Estos valores deben ser incorrectos (algo similar ocurre con los valores demasiado grandes).

Valores atípicos: buenas prácticas

Suele ser un buen hábito repetir el análisis con y sin los valores atípicos, y comparar los resultados.

Si tienen un efecto mínimo en los resultados y no es posible descubrir por qué están en los datos, puede ser razonable reemplazarlos con valores ausentes y seguir adelante con el análisis.

Si tienen un efecto importante en los resultados, no deberían eliminarse sin justificación.

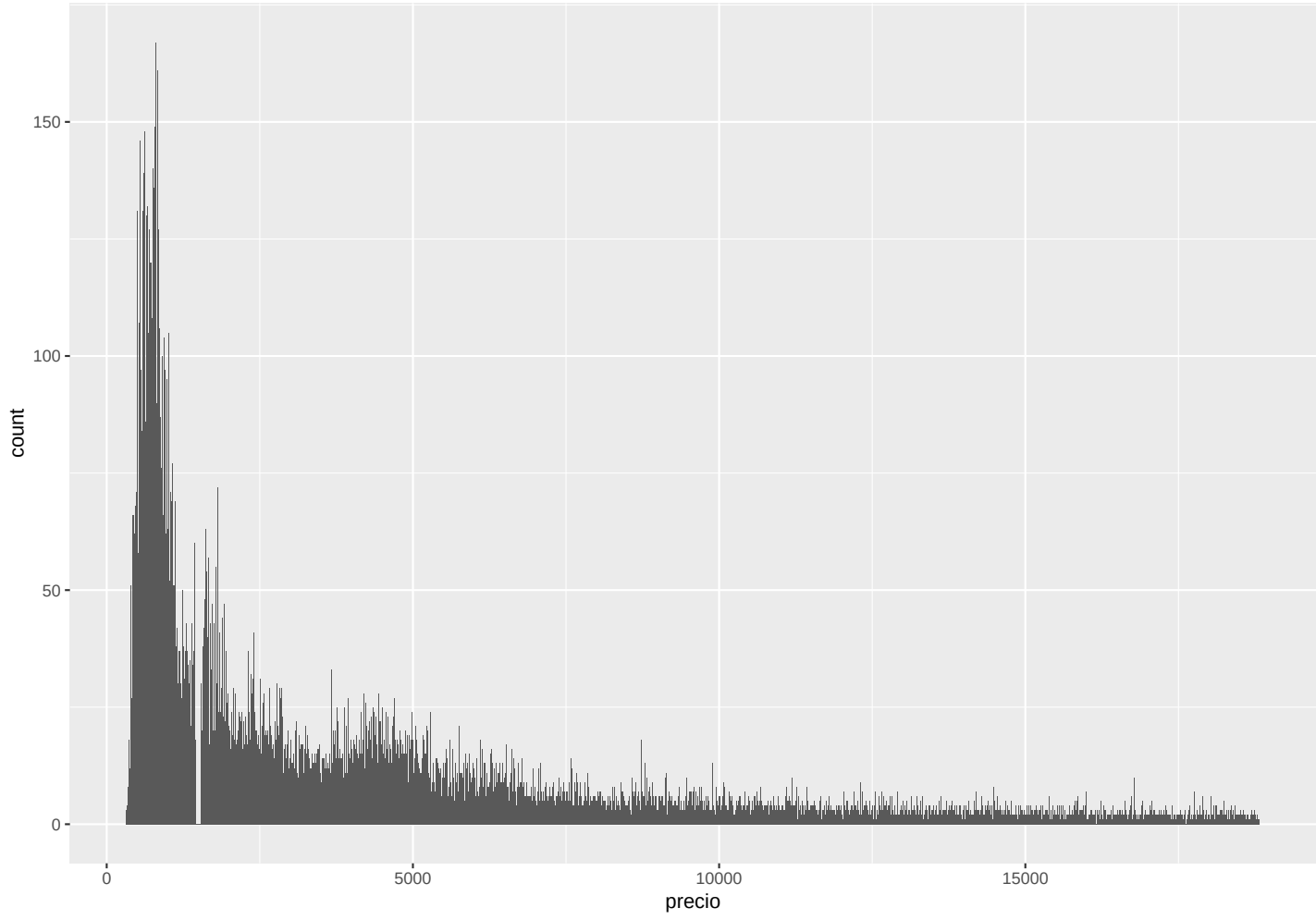
Habrá que descubrir qué los causó (¿error en la entrada de datos?) y explicitar si son removidos en el reporte escrito.

Ejercicios

2. Explora la distribución de precio. ¿Ves algo inusual o sorprendente? (Sugerencia: Piensa detenidamente en `binwidth` y asegúrate de usar un rango largo de valores.)

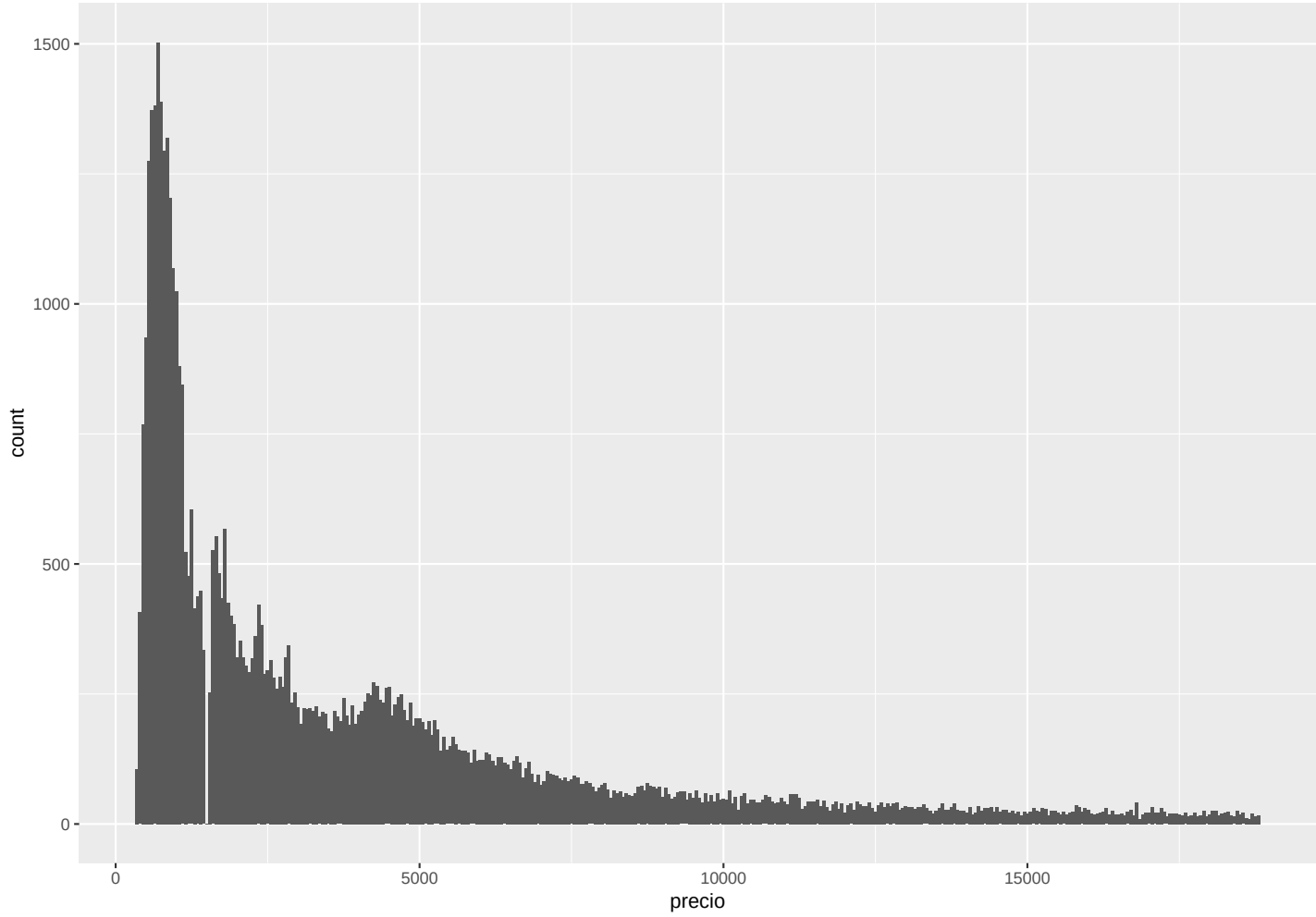
Ejercicio 2 (binwidth = 2)

```
ggplot(diamantes) +
  geom_histogram(mapping = aes(x = precio), binwidth = 2)
```



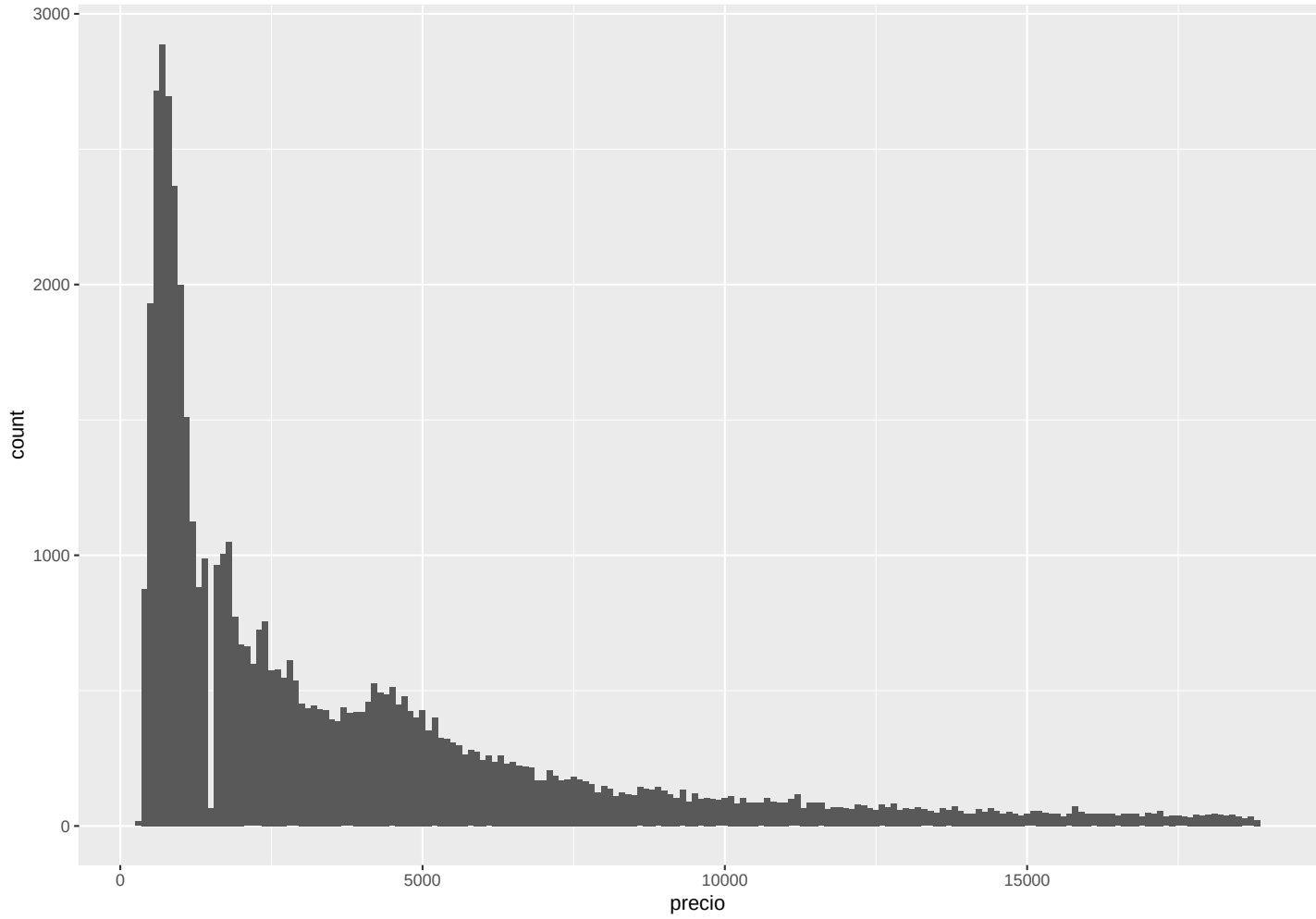
Ejercicio 2 (binwidth = 50)

```
ggplot(diamantes) +
  geom_histogram(mapping = aes(x = precio), binwidth = 50)
```



Ejercicio 2 (binwidth = 100)

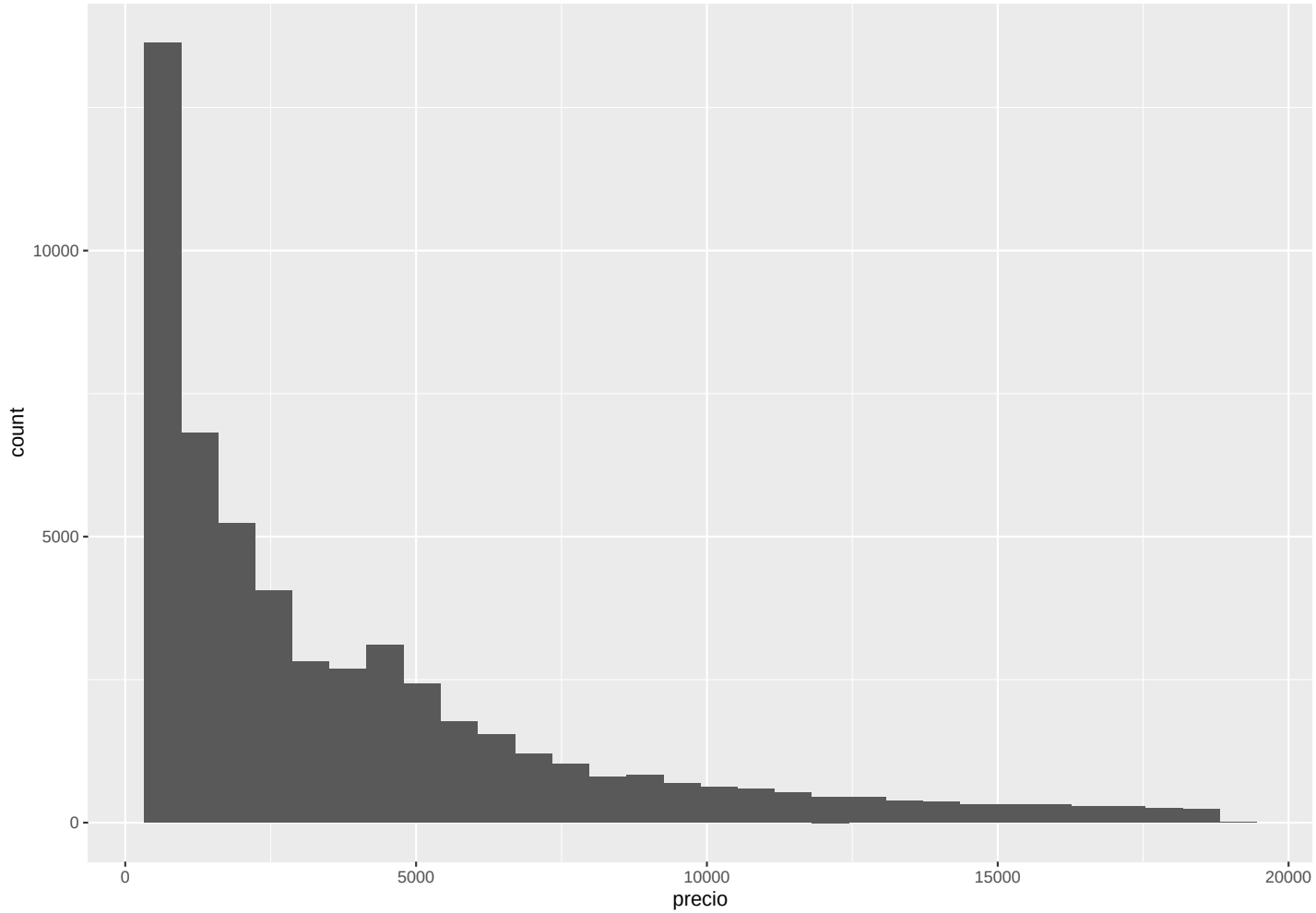
```
ggplot(diamantes) +
  geom_histogram(mapping = aes(x = precio), binwidth = 100)
```



Ejercicio 2 (binwidth por defecto)

```
ggplot(diamantes) +
  geom_histogram(mapping = aes(x = precio))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Valores faltantes (NAs)

Si hay valores atípicos en el conjunto de datos y queremos seguir con el resto del análisis, tenemos dos opciones.

1. Eliminar las filas donde están los valores atípicos (no recomendable):

```
diamantes2 <- diamantes %>%
  filter(between(y, 3, 20))
```

2. Reemplazar los valores inusuales con valores faltantes:

```
diamantes2 <- diamantes %>%
  mutate(y = ifelse(y < 3 | y > 20, NA, y))
```

Valores faltantes (NAs)

Como R en general, ggplot2 sigue la filosofía de que los valores faltantes nunca deberían desaparecer silenciosamente.

La cuestión de dónde graficar los valores faltantes no es trivial, así que ggplot2 no los incluye en los gráficos, pero emite una advertencia acerca de que fueron removidos.

Por ejemplo (gráfico omitido):

```
ggplot(data = diamantes2, mapping = aes(x = x, y = y)) +
  geom_point()
#> Warning: Removed 9 rows containing missing values (geom_point).
```

Ejercicios

2. ¿Qué efecto tiene usar `na.rm = TRUE` en `mean()` (media) y `sum()` (suma)?

```
# "na.rm = T" ignora los NAs y permite hacer los cálculos
mean(c(1:10, NA))
```

```
## [1] NA
```

```
mean(c(1:10, NA), na.rm = T)
```

```
## [1] 5.5
```

```
sum(c(1:10, NA))
```

```
## [1] NA
```

```
sum(c(1:10, NA), na.rm = T)
```

```
## [1] 55
```

Covariación (relaciones entre variables)

Si la variación describe el comportamiento dentro de una variable, la covariación describe el comportamiento entre variables.

La covariación es la tendencia de los valores de dos o más variables a variar simultáneamente de una manera relacionada.

La mejor manera de reconocer que existe covariación en tus datos es visualizar la relación entre dos o más variables y la forma de hacerlo dependerá de los tipos de variables involucradas.

Covariación entre: una variable categórica y una continua

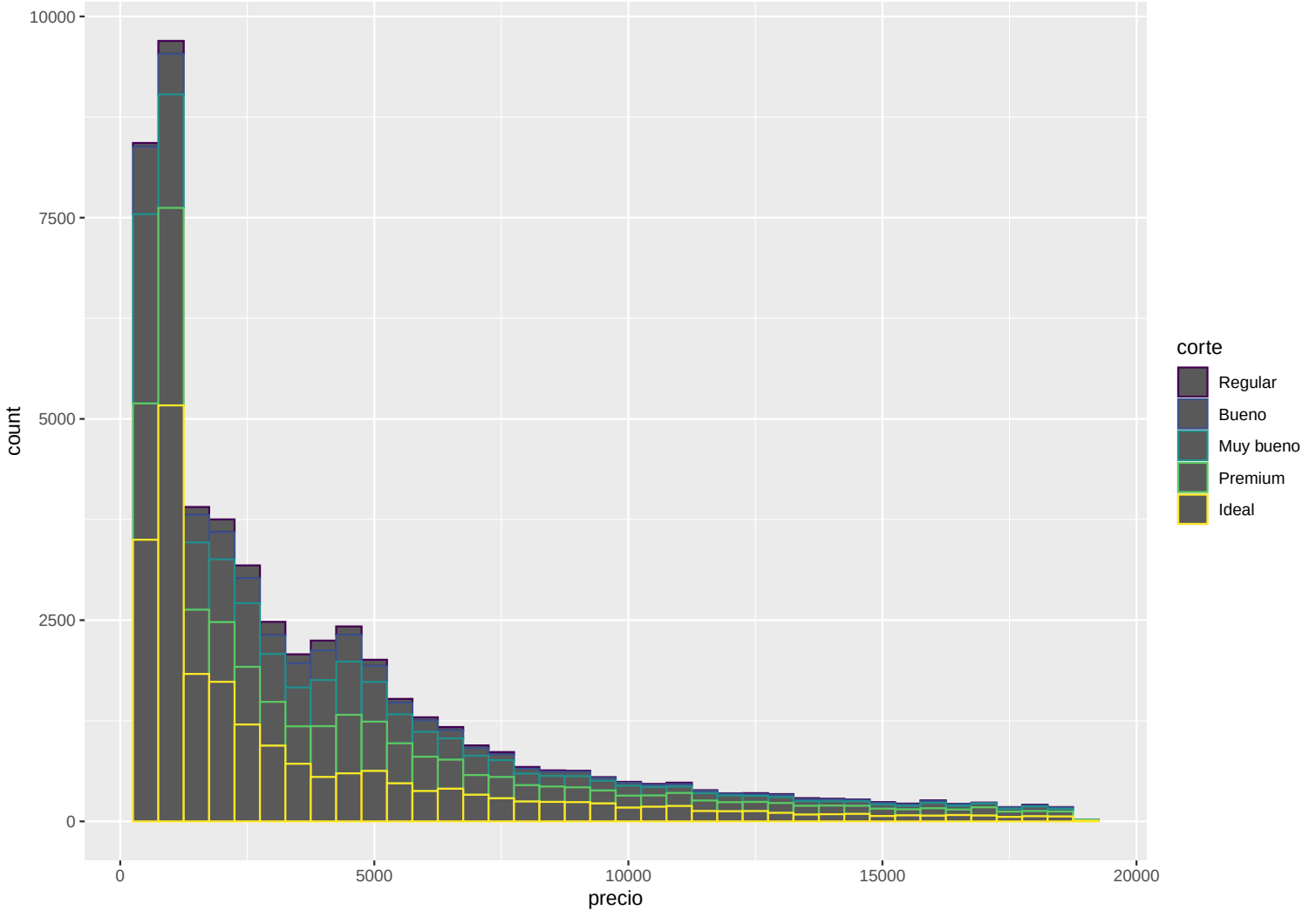
Es frecuente querer explorar la distribución de una variable continua agrupada por una variable categórica.

Una posibilidad consiste en construir varios histogramas de la variable continua, uno para cada categoría de la otra variable.

Pero si deseamos visualizar múltiples distribuciones en la misma gráfica, se recomienda usar `geom_freqpoly` (polígonos de frecuencia) en lugar de `geom_histogram`, que usa líneas en lugar de barras para mostrar los totales.

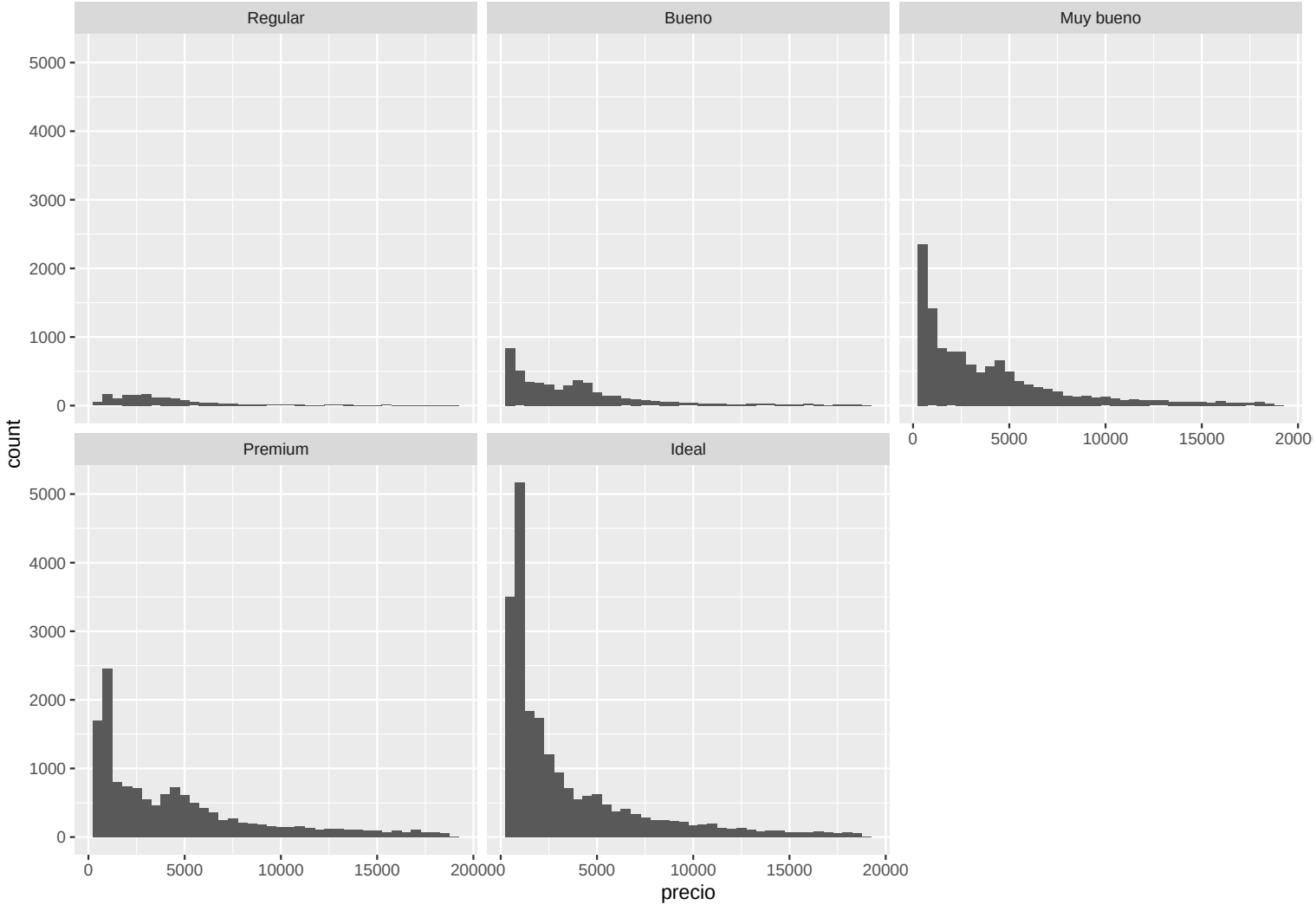
Ejemplo: precio vs. corte (histogramas superpuestos)

```
ggplot(data = diamantes, mapping = aes(x = precio, colour = corte)) +
  geom_histogram(binwidth = 500)
```



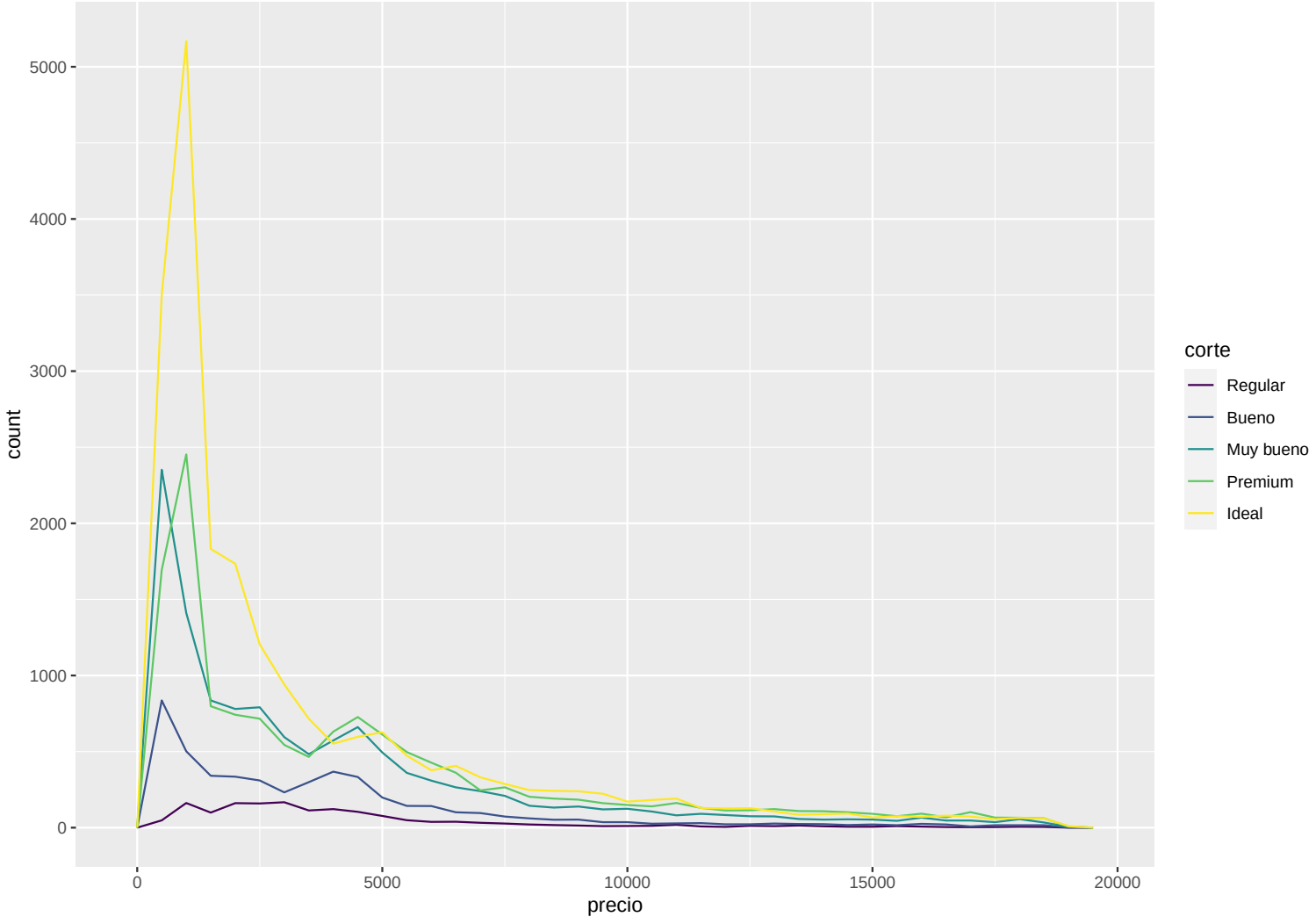
Ejemplo: precio vs. corte (múltiples histogramas)

```
ggplot(data = diamantes, mapping = aes(x = precio)) +
  geom_histogram(binwidth = 500) + facet_wrap( ~ corte)
```



Ejemplo: precio vs. corte (polígonos de frecuencia)

```
ggplot(data = diamantes, mapping = aes(x = precio, colour = corte)) +
  geom_freqpoly(binwidth = 500)
```



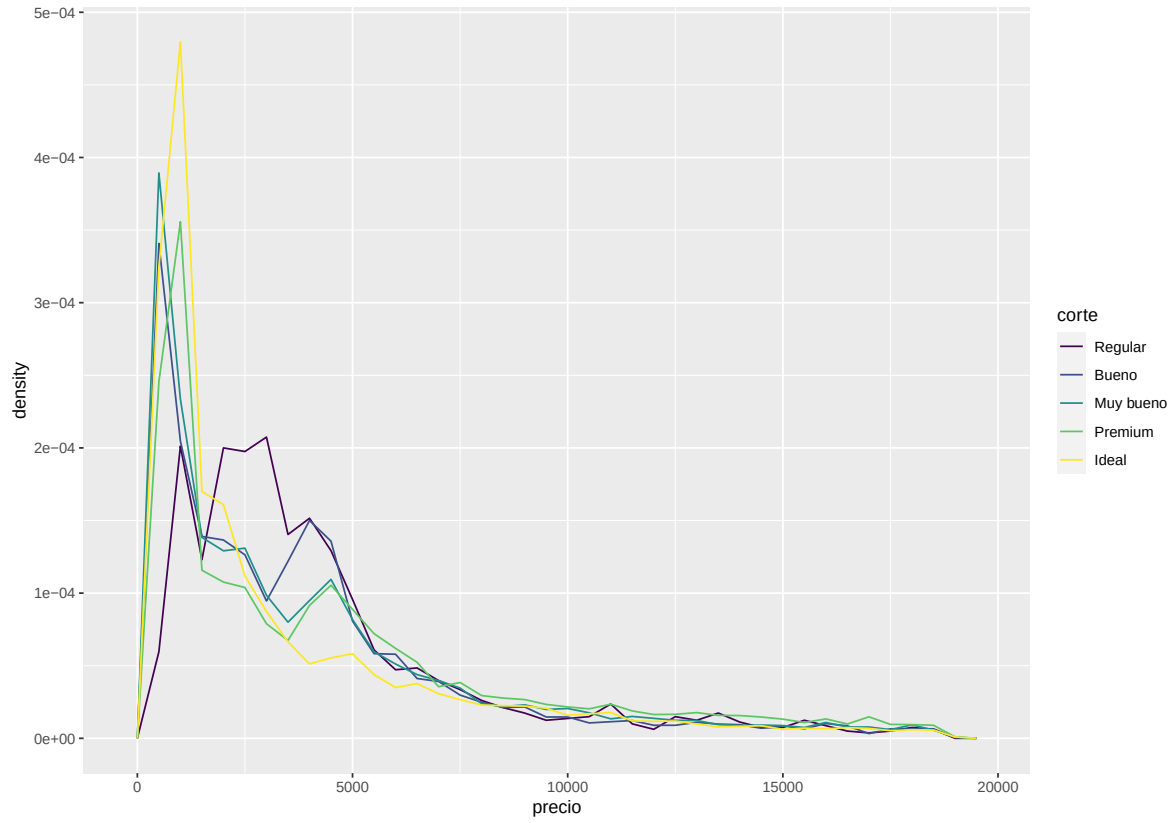
Ejemplo: relación entre `precio` y `corte`

Resulta difícil observar las diferencias de las distribuciones porque las frecuencias totales difieren mucho entre categorías (`Regular` 1610, `Bueno` 4906, `Muy bueno` 12082, `Premium` 13791, `Ideal` 21551).

Para facilitar esta comparación se cambia lo que se muestra en el eje vertical (frecuencia absoluta por densidad), de manera que el área bajo cada polígono es igual a 1.

Ejemplo: precio vs. corte (polígonos de densidad)

```
ggplot(data = diamantes, mapping = aes(x = precio, y = ..density..)) +
  geom_freqpoly(mapping = aes(colour = corte), binwidth = 500)
```



Observación: ¡parece que los diamantes regulares (la calidad más baja) tienen, en promedio, el precio más alto!

Diagramas de Caja (*boxplots*)

Una alternativa para mostrar la distribución de una variable continua agrupada en relación con otra variable categórica es el diagrama de caja (boxplot en inglés).

Cada diagrama de caja está integrado por:

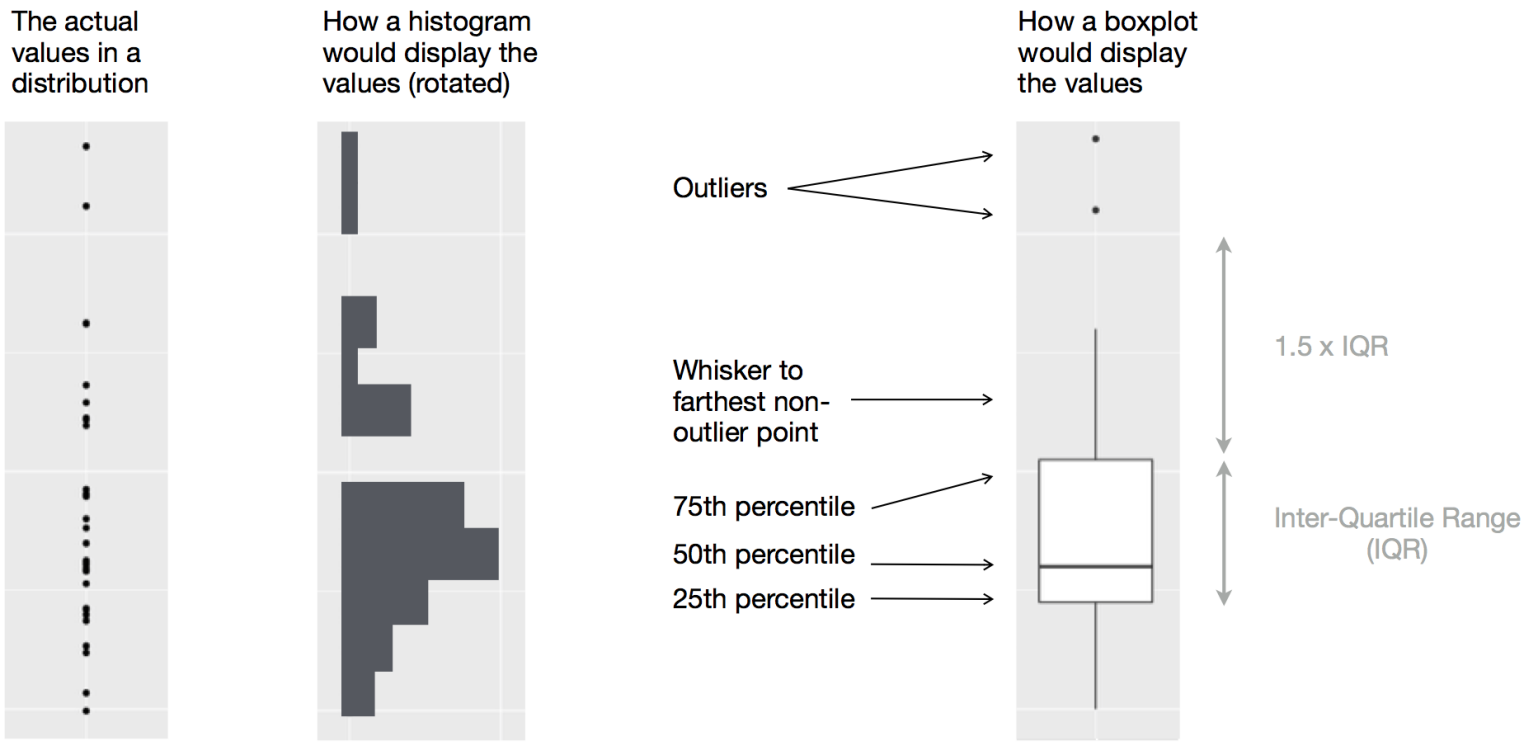
(i) una caja que va desde el percentil 25 al 75 (la diferencia entre estos números es el rango intercuartílico o *IQR*);

(ii) una línea que señala la mediana (percentil 50);

(iii) puntos que representan observaciones que se encuentran a más de 1.5 veces el IQR a partir de cualquier borde de la caja (considerados atípicos);

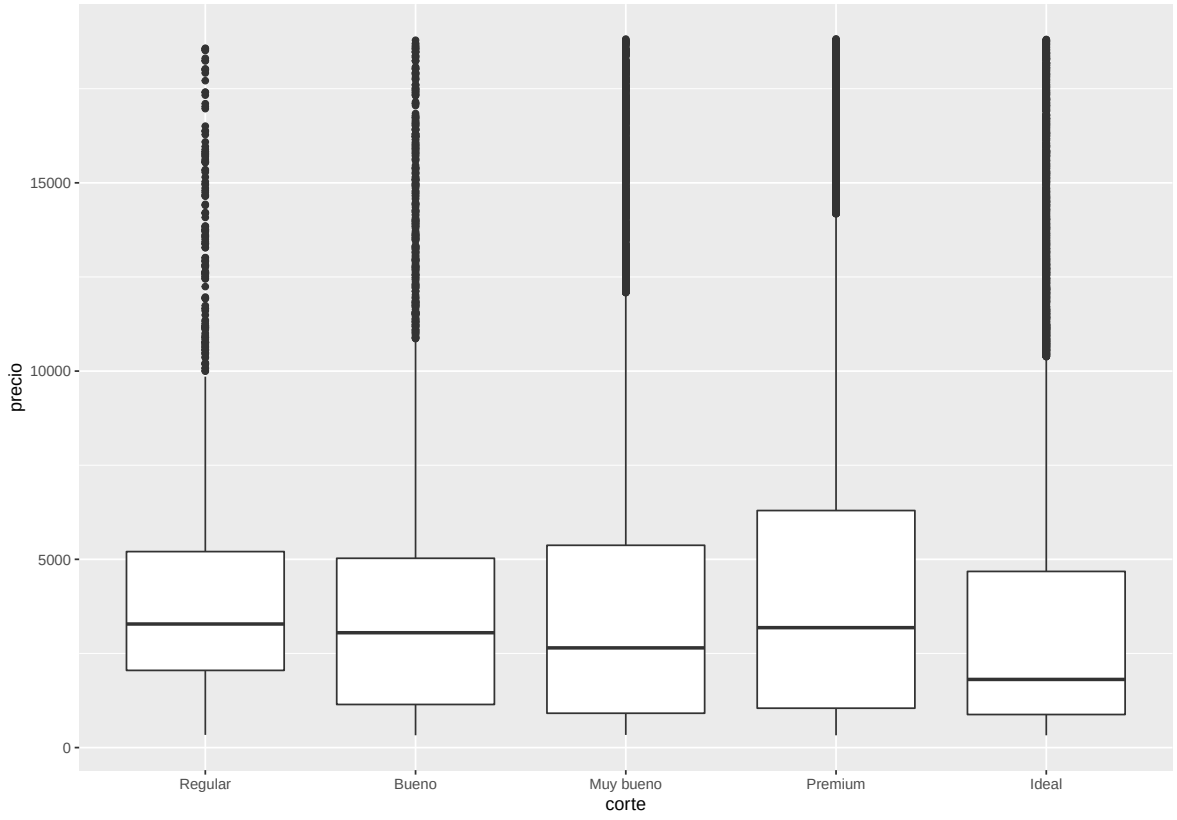
(iv) una línea (o bigote) desde cada borde de la caja hasta el punto más lejano de la distribución que no sea considerado atípico.

Diagrama de Caja



Ejemplo: ... precio y corte (diagramas de caja)

```
ggplot(data = diamantes, mapping = aes(x = corte, y = precio)) +
  geom_boxplot()
```



Observación: si bien vemos menos información sobre la distribución, son más compactos y por eso resulta más fácil compararlos.

Reordenando variables categóricas

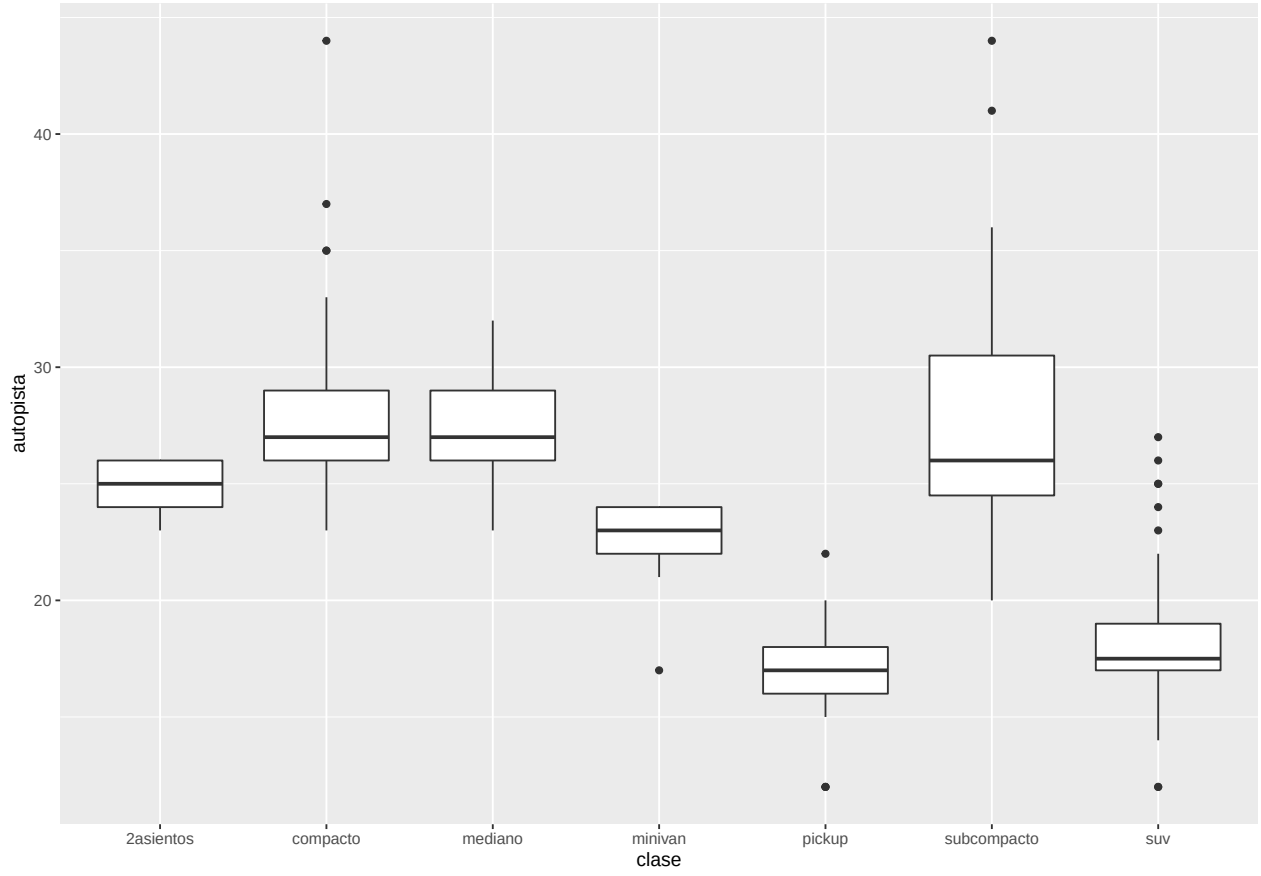
En el ejempolo anterior la variable `corte` es un factor ordenado.

Otras variables categóricas no tienen un orden intrínseco, de manera que para graficarlas puede convenir reordenarlas de acuerdo a la variable continua de interés y de esa manera observar mejor la tendencia.

Una manera de hacerlo es usando la función `reorder`.

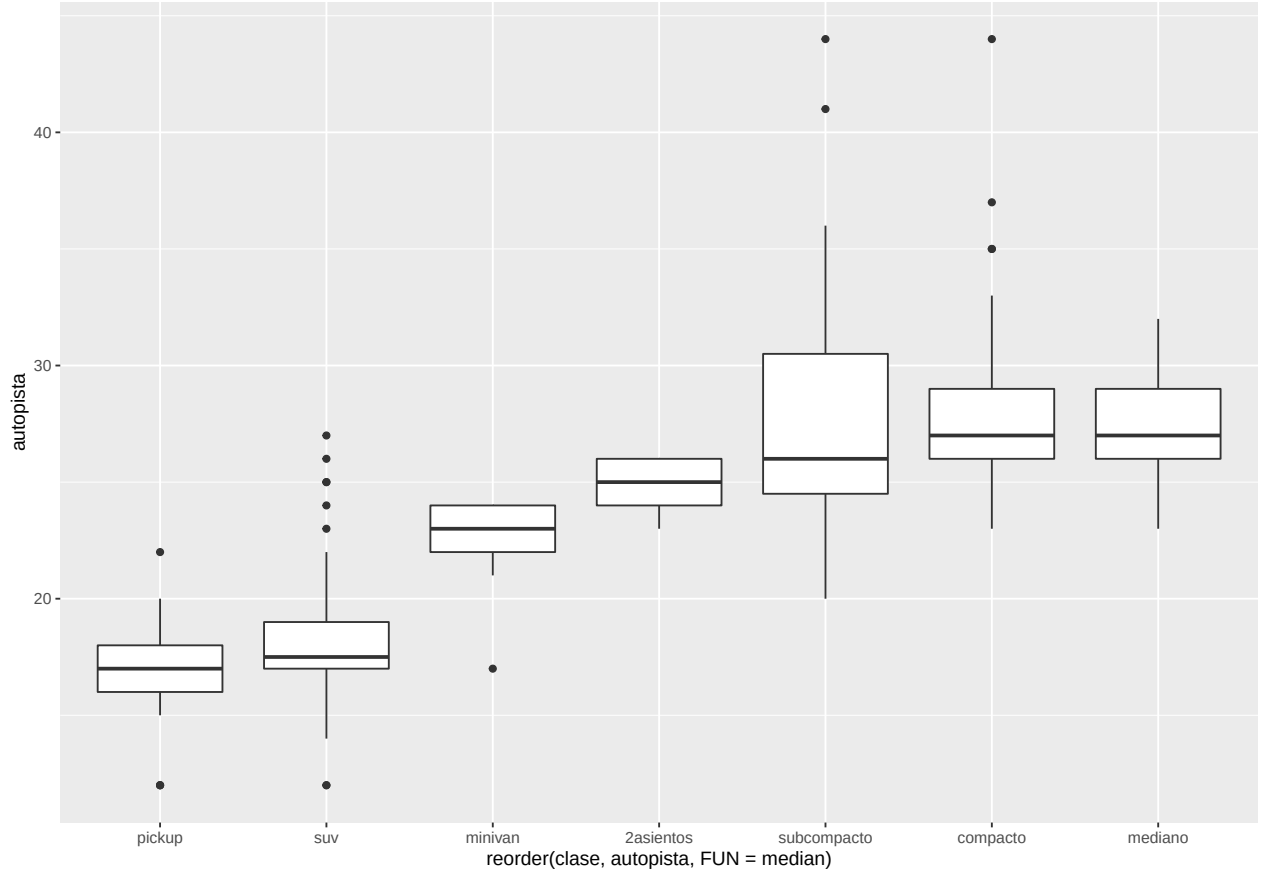
Ejemplo: factor no ordenado (datos `millas`)

```
ggplot(data = millas, mapping = aes(x = clase, y = autopista)) +
  geom_boxplot()
```



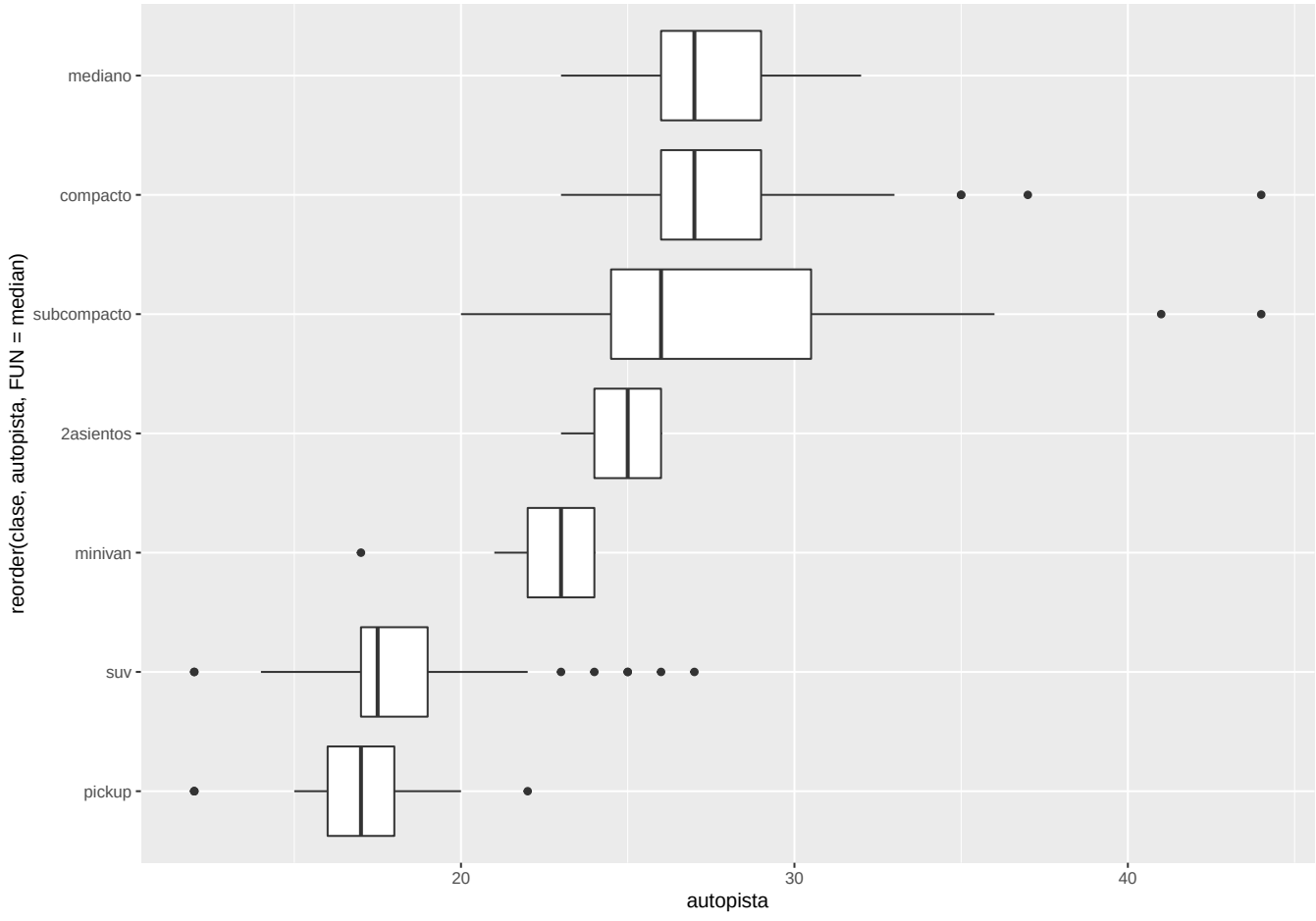
Ejemplo: factor reordenado (respecto a la mediana de autopista)

```
ggplot(data = millas, mapping = aes(x =
                reorder(clase, autopista, FUN = median), y = autopista)) +
  geom_boxplot()
```



Ejemplo: giro de 90° con `coord_flip` para etiquetas largas

```
ggplot(data = millas) +
  geom_boxplot(mapping = aes(x = reorder(clase, autopista, FUN = median),
                             y = autopista)) +
  coord_flip()
```



Ejercicios

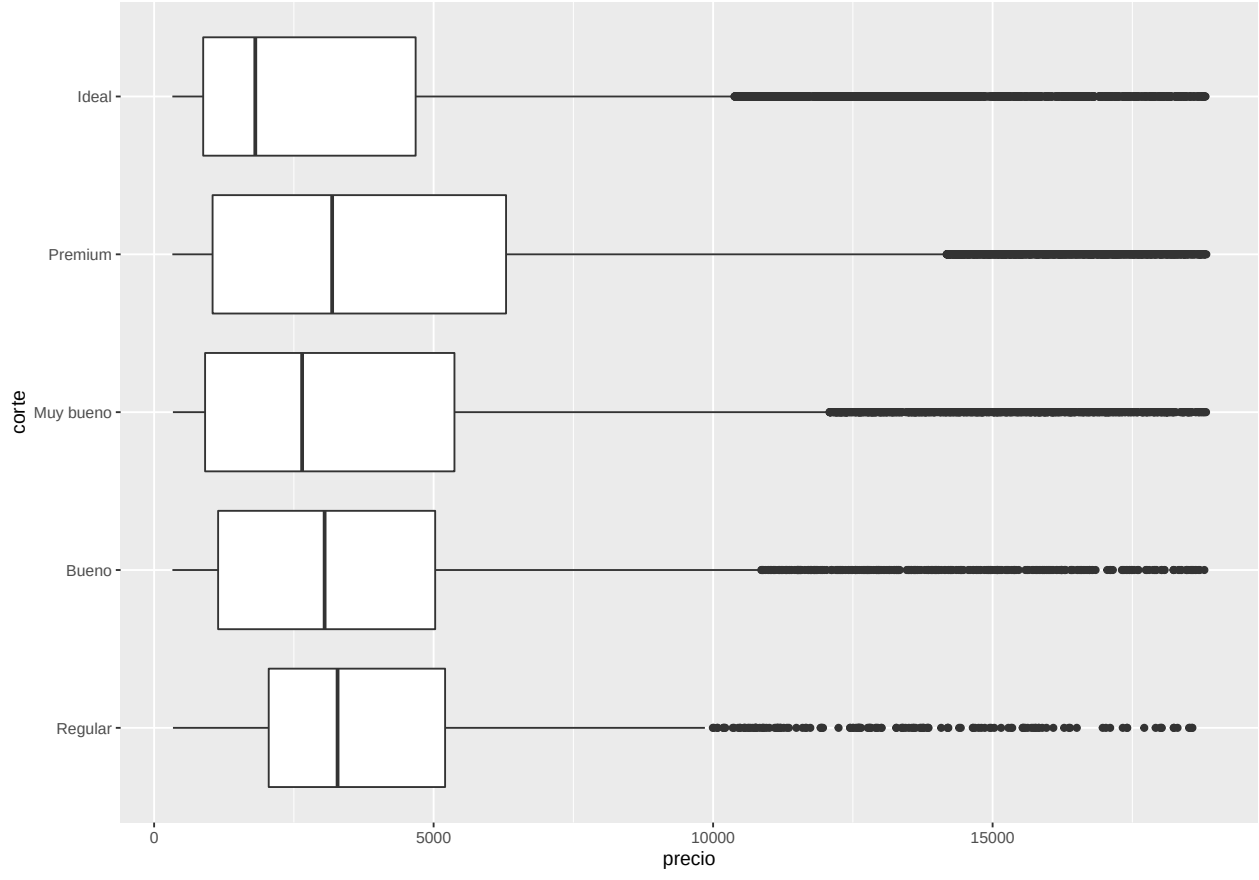
2. ¿Qué variable del conjunto de datos de diamantes es más importante para predecir el precio de un diamante?

   ¿Cómo está correlacionada esta variable con el corte?

   ¿Por qué la combinación de estas dos relaciones conlleva que los diamantes de menor calidad sean más costosos?

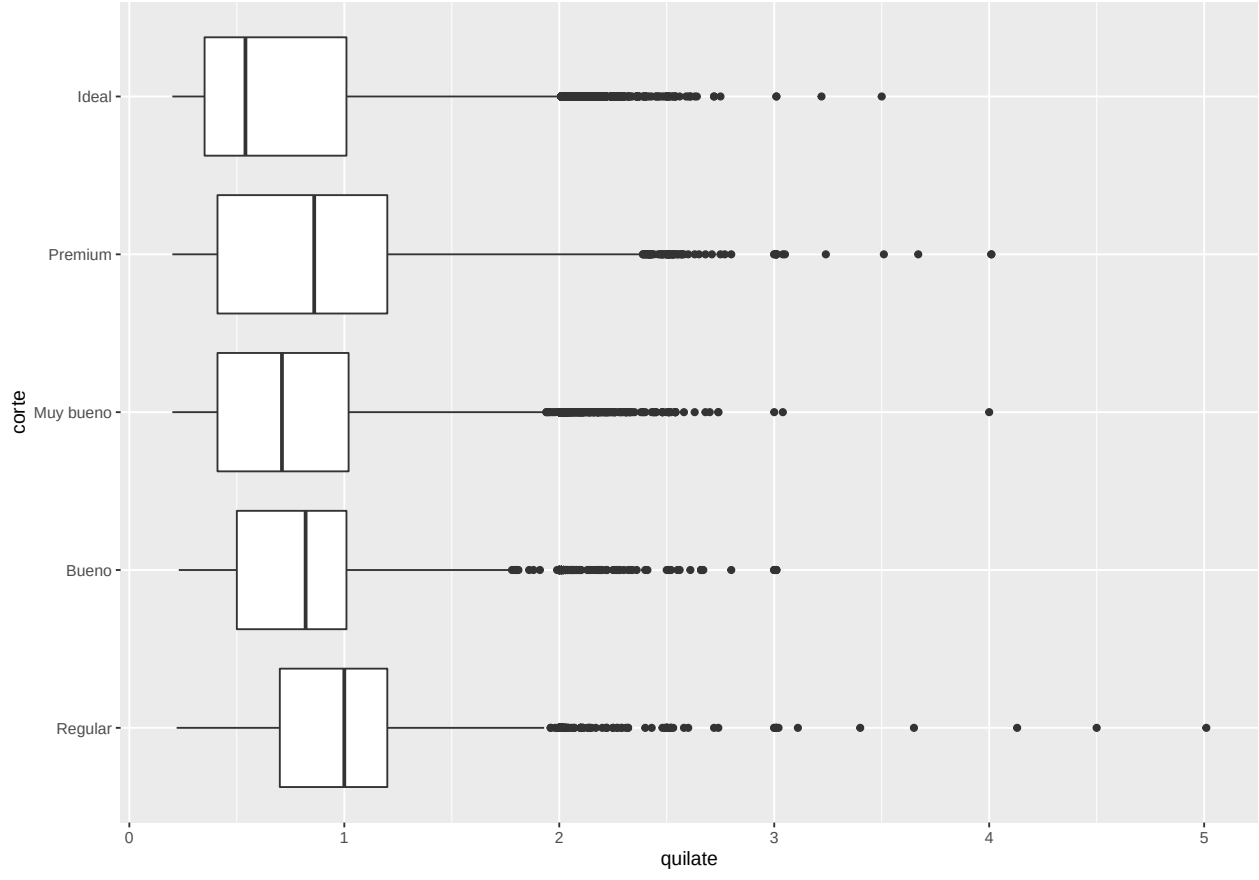
Covariación entre corte y precio

```
ggplot(data = diamantes) +
  geom_boxplot(mapping = aes(x = corte, y = precio)) +
  coord_flip()
```



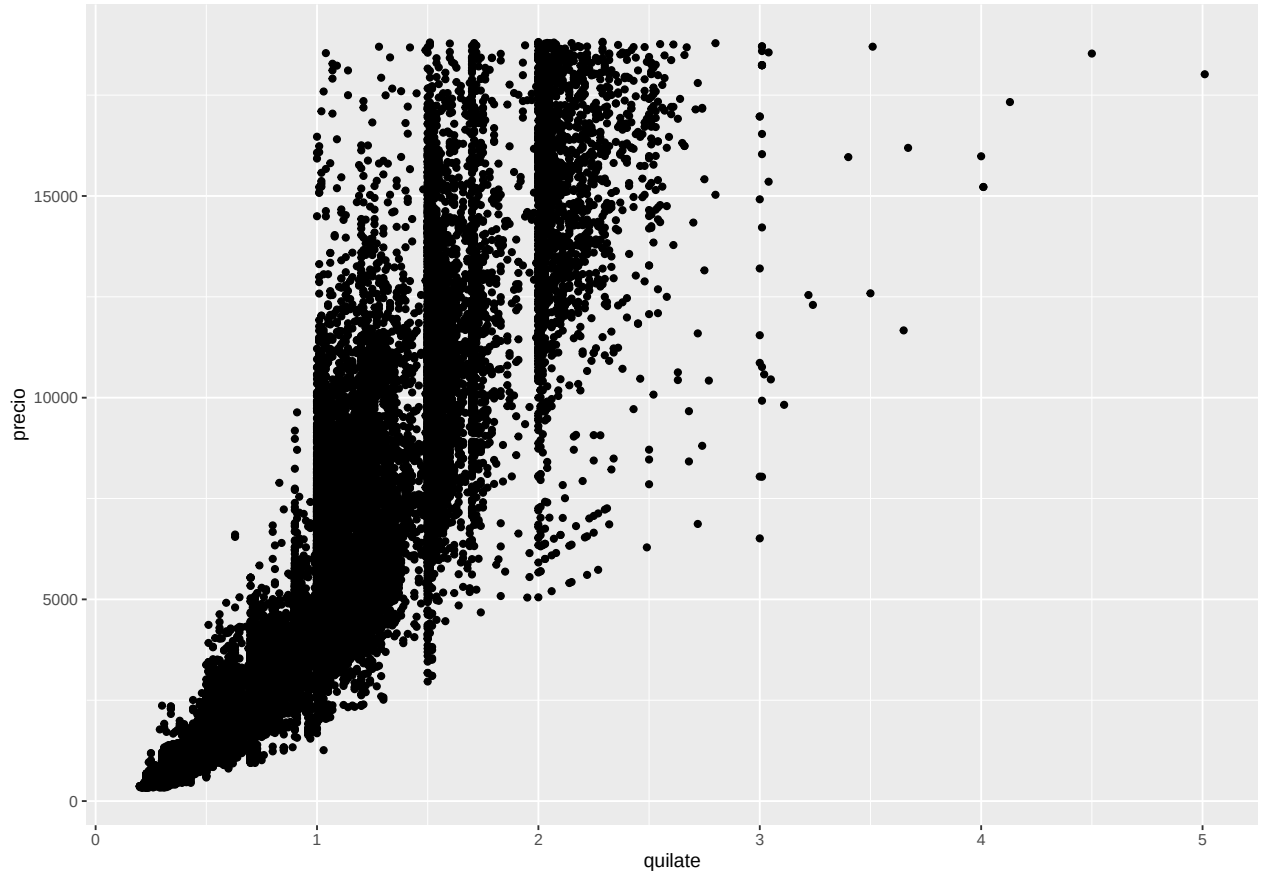
Covariación entre corte y quilate

```
ggplot(data = diamantes) +
  geom_boxplot(mapping = aes(x = corte, y = quilate)) +
  coord_flip()
```



Covariación entre quilate y precio

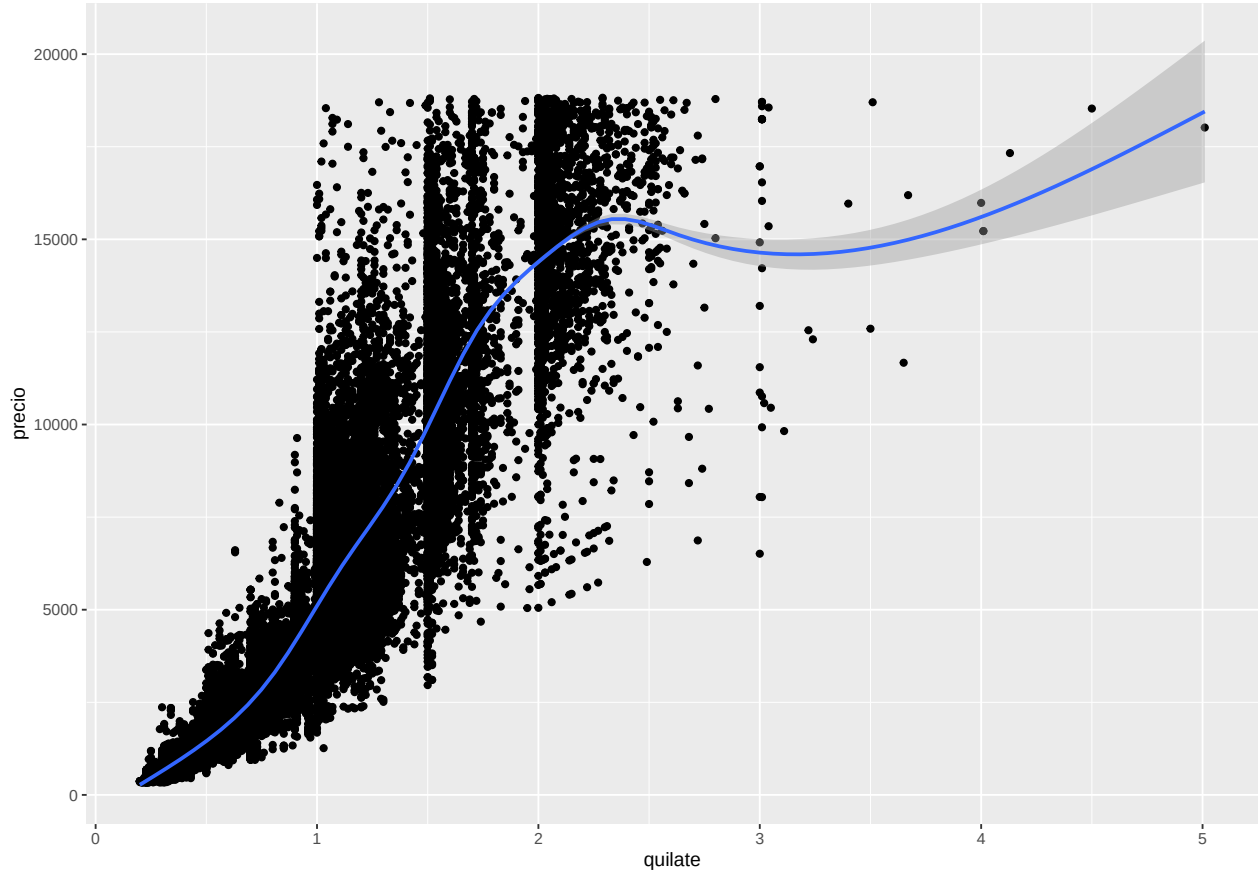
```
ggplot(data = diamantes) +
  geom_point(mapping = aes(x = quilate, y = precio))
```



Patrón de covariación entre quilate y precio

```
ggplot(data = diamantes) +
  geom_point(mapping = aes(x = quilate, y = precio)) +
  geom_smooth(mapping = aes(x = quilate, y = precio))
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Algunas conclusiones

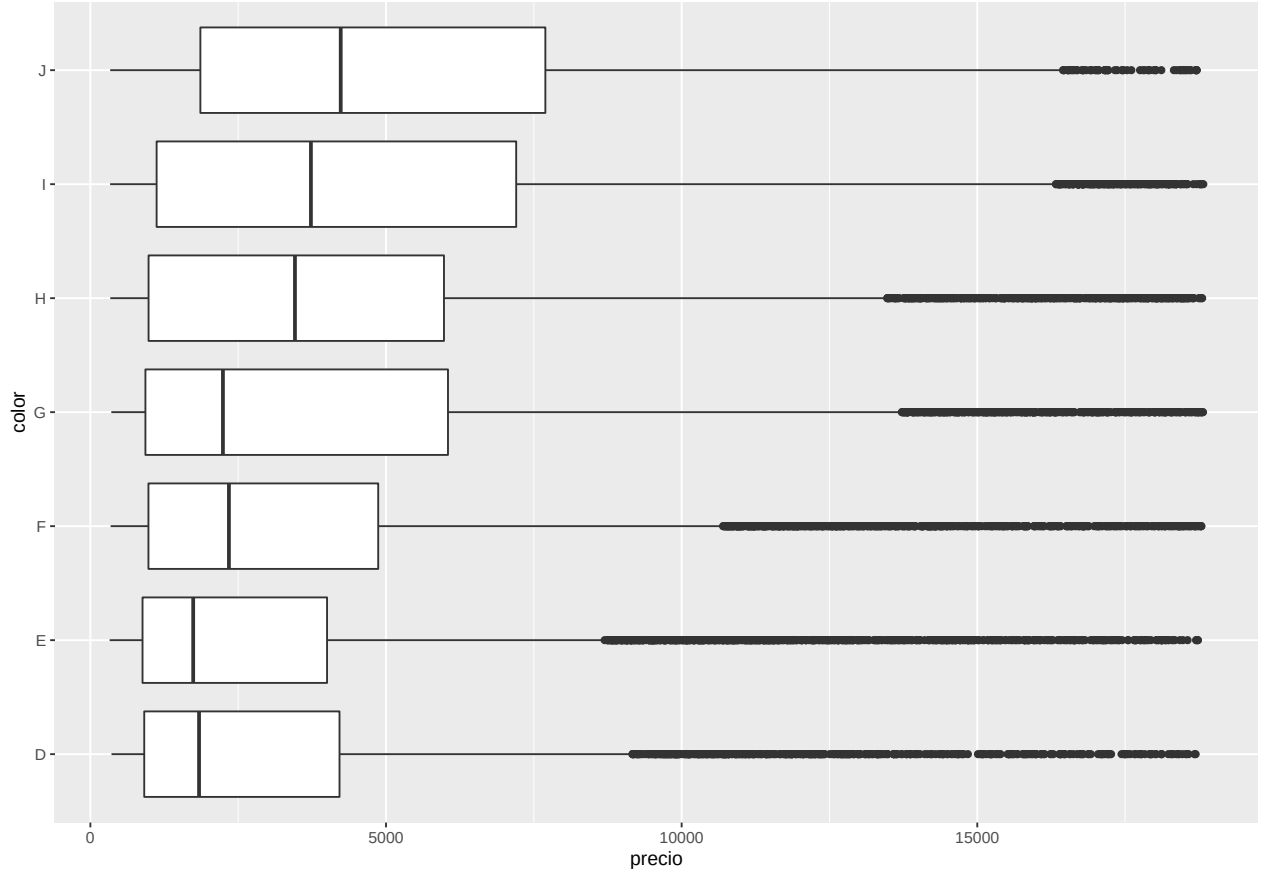
La variable `quilate` parece ser importante para explicar o predecir el precio de un diamante.

En definitiva, la explicación de por qué peores cortes tiene en promedio mayores precios se debe a que peores cortes tienen, en promedio, mayor quilate y, por tanto, mayor precio (promedio).

Otras variables presentes en la base están también relacionadas con el precio. Por ejemplo, `color` y `claridad`.

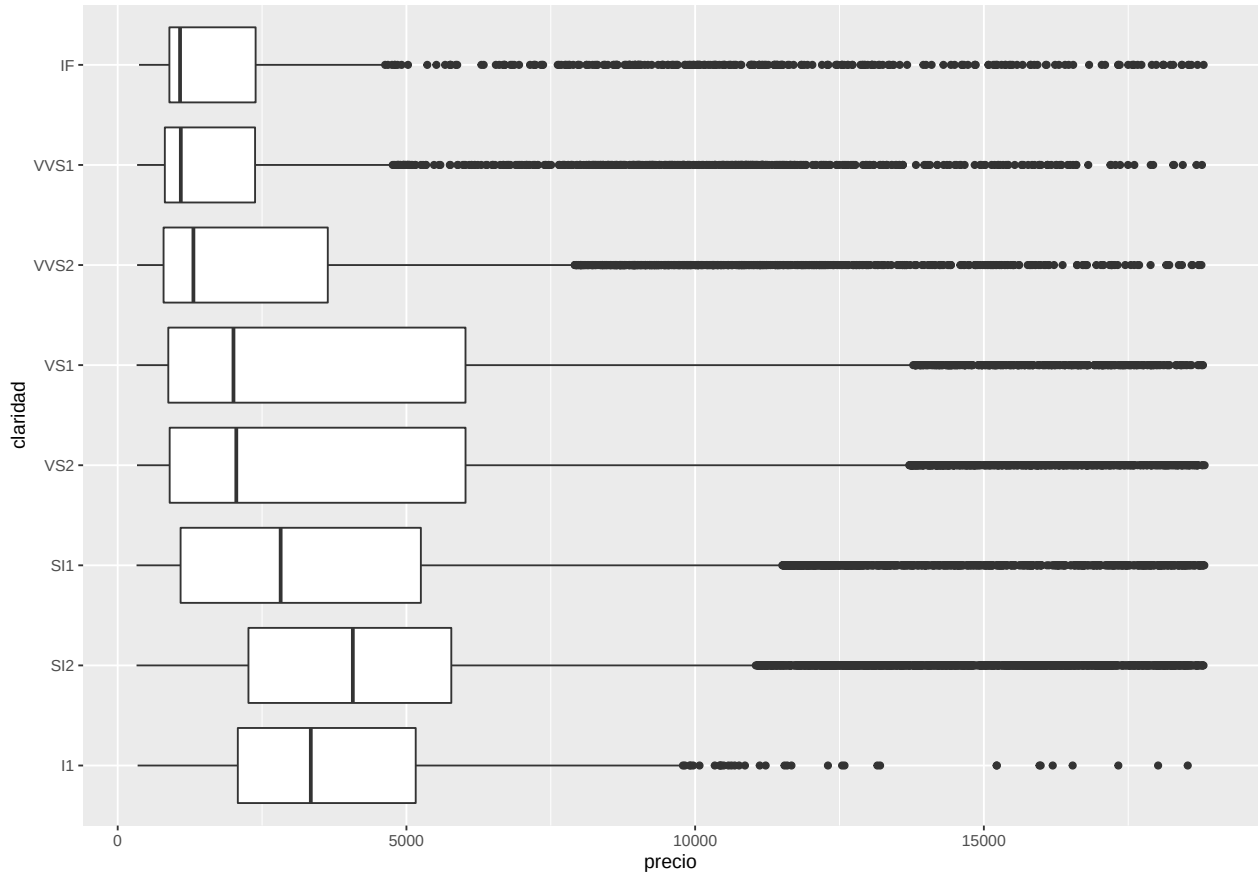
Covariación entre color y precio

```
ggplot(data = diamantes) +
  geom_boxplot(mapping = aes(x = color, y = precio)) +
  coord_flip()
```



Covariación entre claridad y precio

```
ggplot(data = diamantes) +
  geom_boxplot(mapping = aes(x = claridad, y = precio)) +
  coord_flip()
```



(fin del ejercicio)

Covariación entre: dos variables categóricas

Para analizar la covariación entre variables categóricas, debemos contar el número de observaciones de cada combinación.

Una manera de hacerlo es empleando la función `count`. Por ejemplo, para las variables `color` y `corte` de `diamantes`:

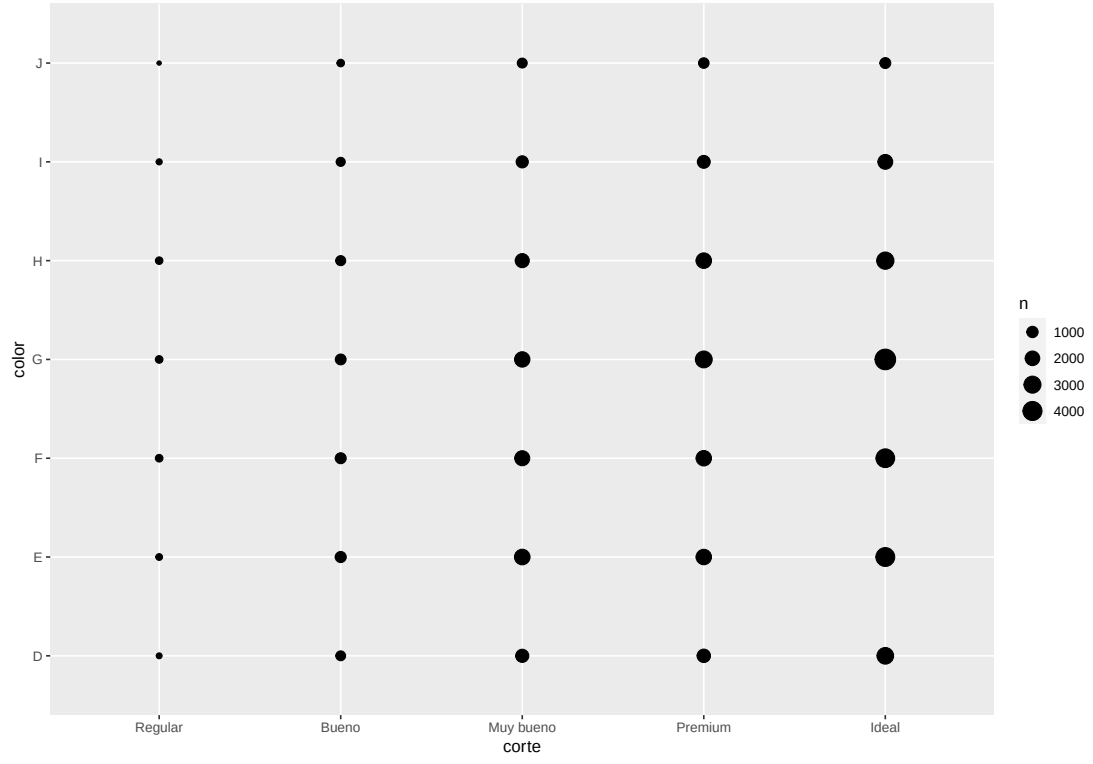
```
diamantes %>%
  count(color, corte)
```

```
## # A tibble: 35 x 3
##    color corte          n
##    <ord> <ord>      <int>
##  1 D     Regular      163
##  2 D     Bueno        662
##  3 D     Muy bueno   1513
##  4 D     Premium     1603
##  5 D     Ideal       2834
##  6 E     Regular      224
##  7 E     Bueno        933
##  8 E     Muy bueno   2400
##  9 E     Premium     2337
## 10 E     Ideal       3903
## # ... with 25 more rows
```

Ejemplo: covariación entre `color` y `corte`

Otra es graficando estas frecuencias con la función `geom_count` donde el tamaño de cada círculo en la gráfica muestra cuántas observaciones corresponden a cada combinación de valores.

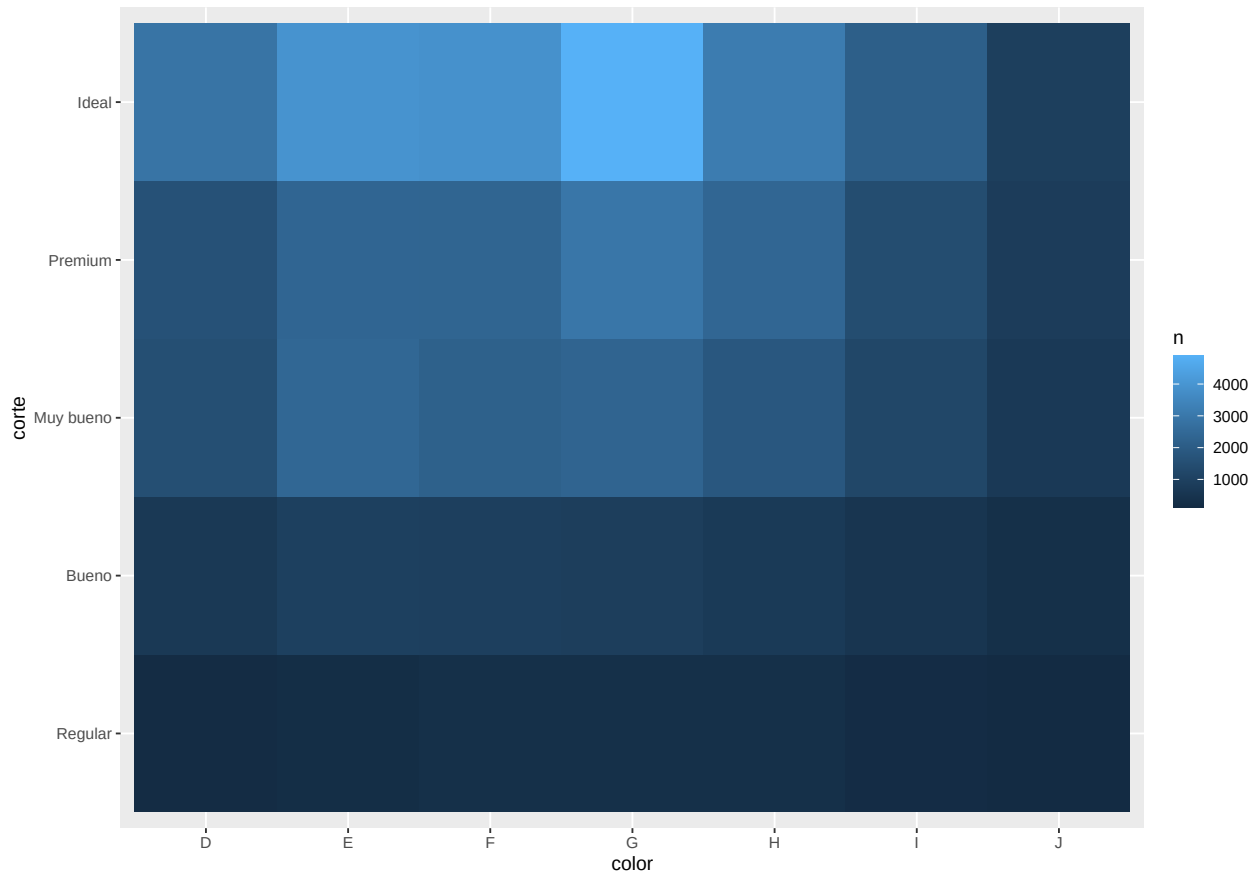
```
ggplot(data = diamantes) +
  geom_count(mapping = aes(x = corte, y = color))
```



Ejemplo: covariación entre `color` y `corte`

Otra alternativa gráfica consiste en la función `geom_tile` y adaptar la estética de relleno (fill):

```
diamantes %>%
  count(color, corte) %>%
  ggplot(mapping = aes(x = color, y = corte)) +
    geom_tile(mapping = aes(fill = n))
```

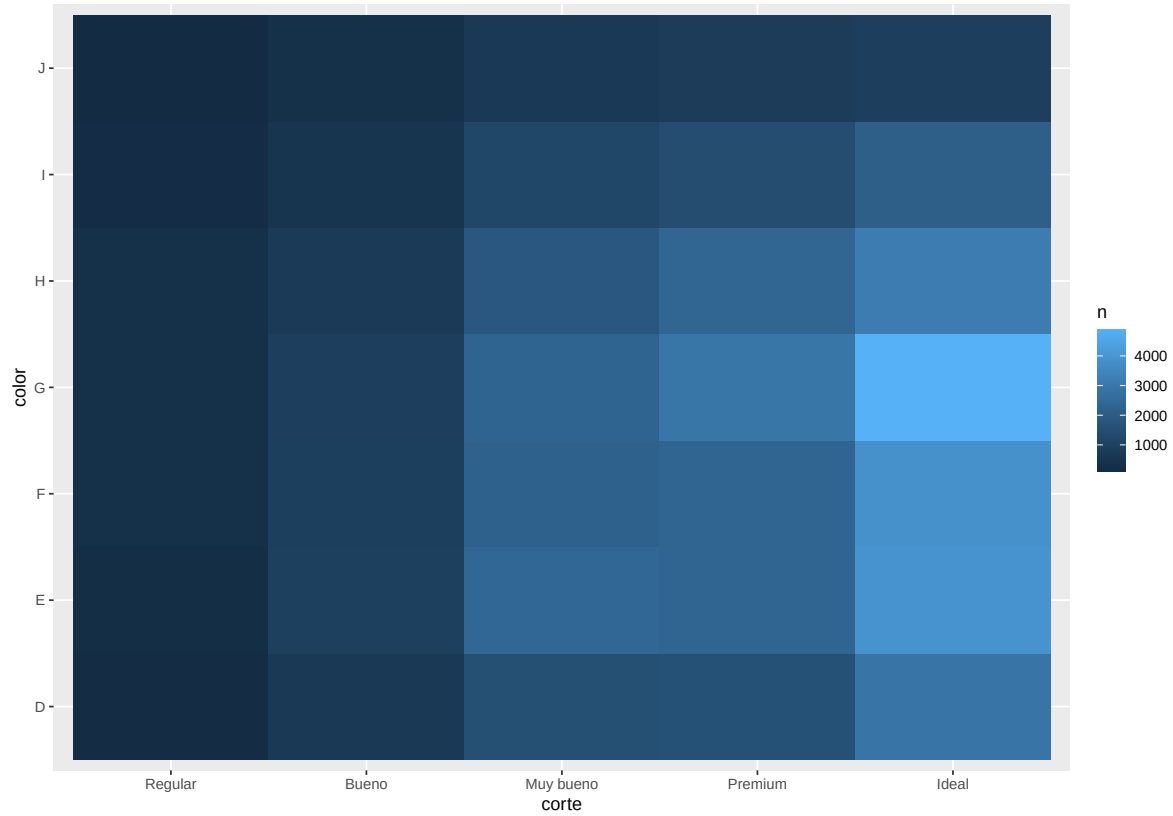


Ejercicios

3. ¿Por qué es un poco mejor usar `aes(x = color, y = corte)` en lugar de `aes(x = corte, y = color)` en el ejemplo anterior?

Ejercicios

```
diamantes %>%
  count(corte, color) %>%
  ggplot(mapping = aes(x = corte, y = color)) +
  geom_tile(mapping = aes(fill = n))
```



Respuesta: ¿por el largo de las etiquetas en `corte`?

Covariación entre: dos variables continuas

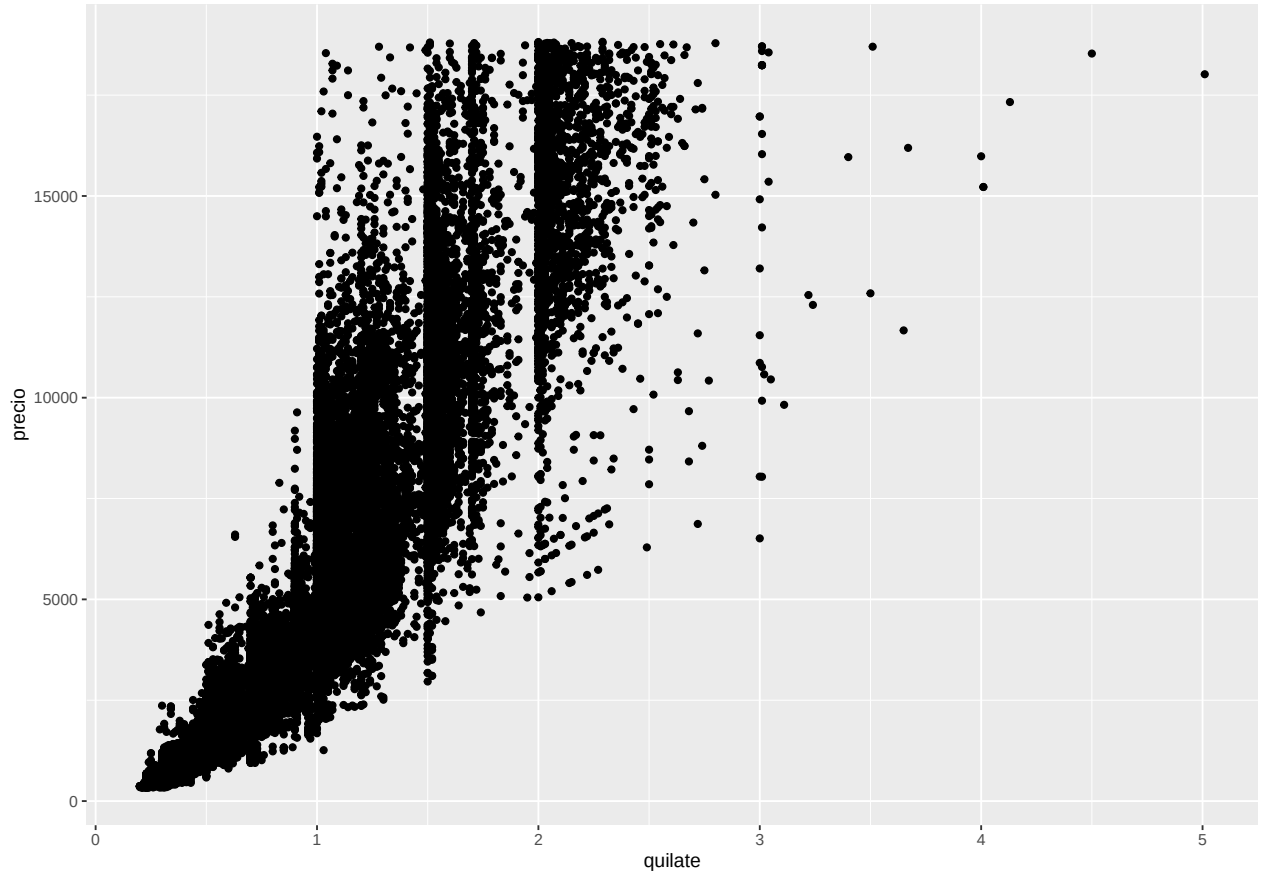
Ya vimos una manera muy buena (y habitual) de visualizar la covariación entre dos variables continuas: dibujar un diagrama de dispersión (*scatterplot* en inglés) con `geom_point`.

En este gráfico es posible identificar la covariación como un patrón en la distribución de los puntos.

Por ejemplo, el siguiente gráfico sugiere una relación casi exponencial entre los quilates y el precio de los diamantes.

Ejemplo: covariación entre `quilate` y `precio`

```
ggplot(data = diamantes) +
  geom_point(mapping = aes(x = quilate, y = precio))
```



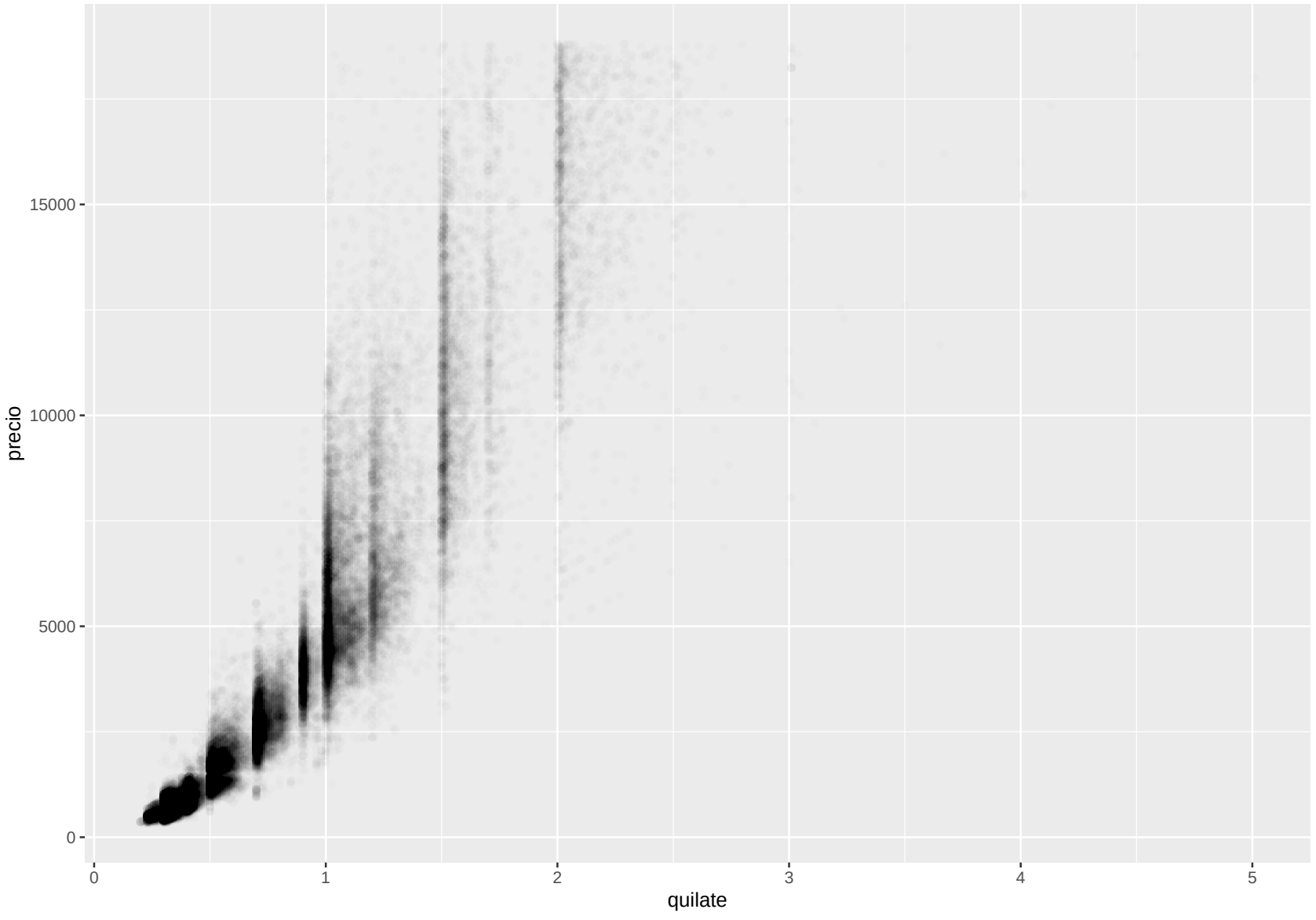
Mejoras sobre el diagrama de dispersión: transparencia

Los diagramas de dispersión resultan menos útiles cuando el conjunto de datos es muy grande y hay valores similares, porque los puntos se superponen (como en el ejemplo anterior).

Una manera de solucionar este problema es usando el parámetro `alpha` para agregar transparencia a los puntos.

```
ggplot(data = diamantes) +
  geom_point(mapping = aes(x = quilate, y = precio), alpha = 1 / 100)
```

Diagrama de dispersión con transparencia



Mejoras sobre el diagrama de dispersión: a colorear

Otra solución es usar las funciones `geom_bin2d` o `geom_hex` para segmentar las variables en rangos.

Estas funciones dividen el plano cartesiano en unidades o contenedores bidimensionales y luego usan un color de relleno para mostrar cuántos puntos pertenecen a cada contenedor.

La función `geom_bin2d` crea contenedores rectangulares mientras que `geom_hex` crea contenedores hexagonales.

Diagrama de dispersión con contenedores rectangulares

```
ggplot(data = diamantes) +
  geom_bin2d(mapping = aes(x = quilate, y = precio))
```

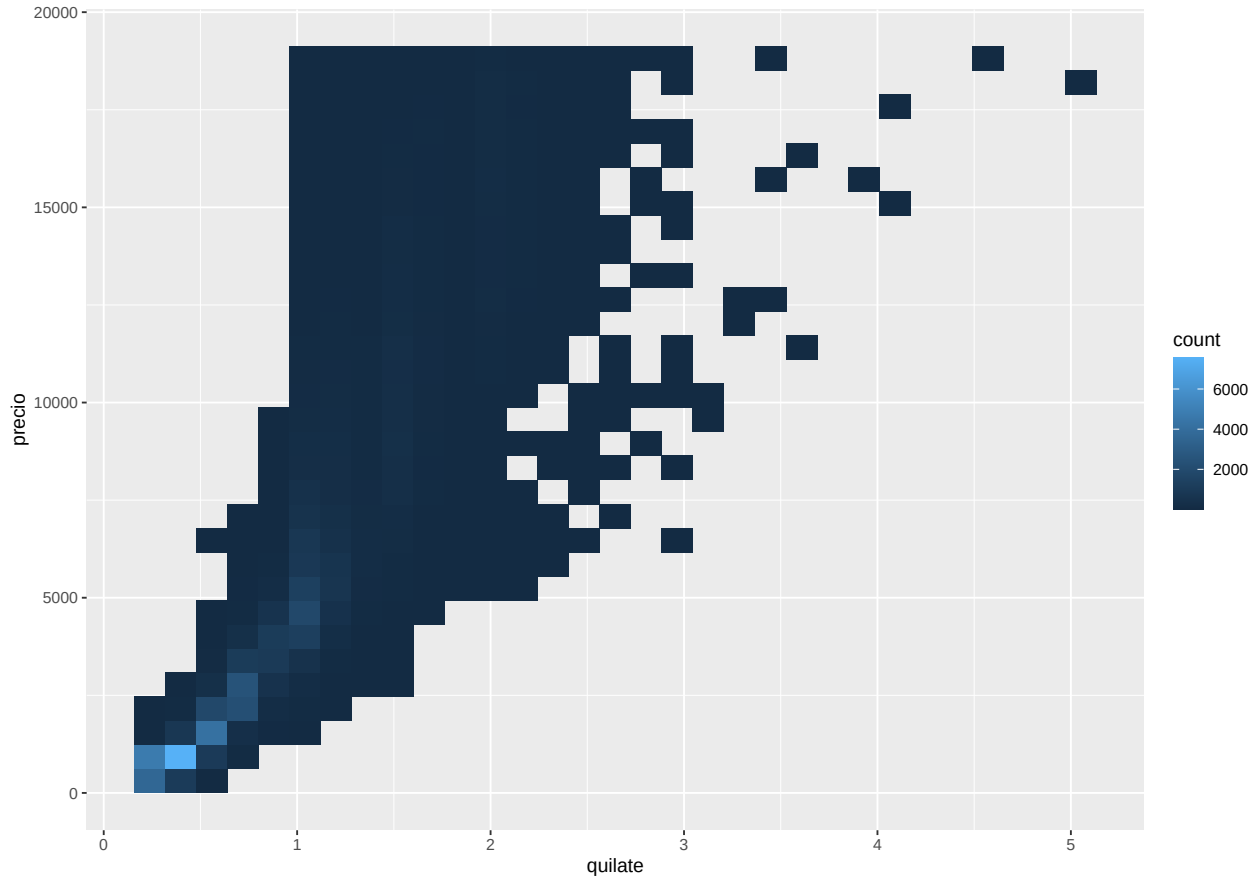
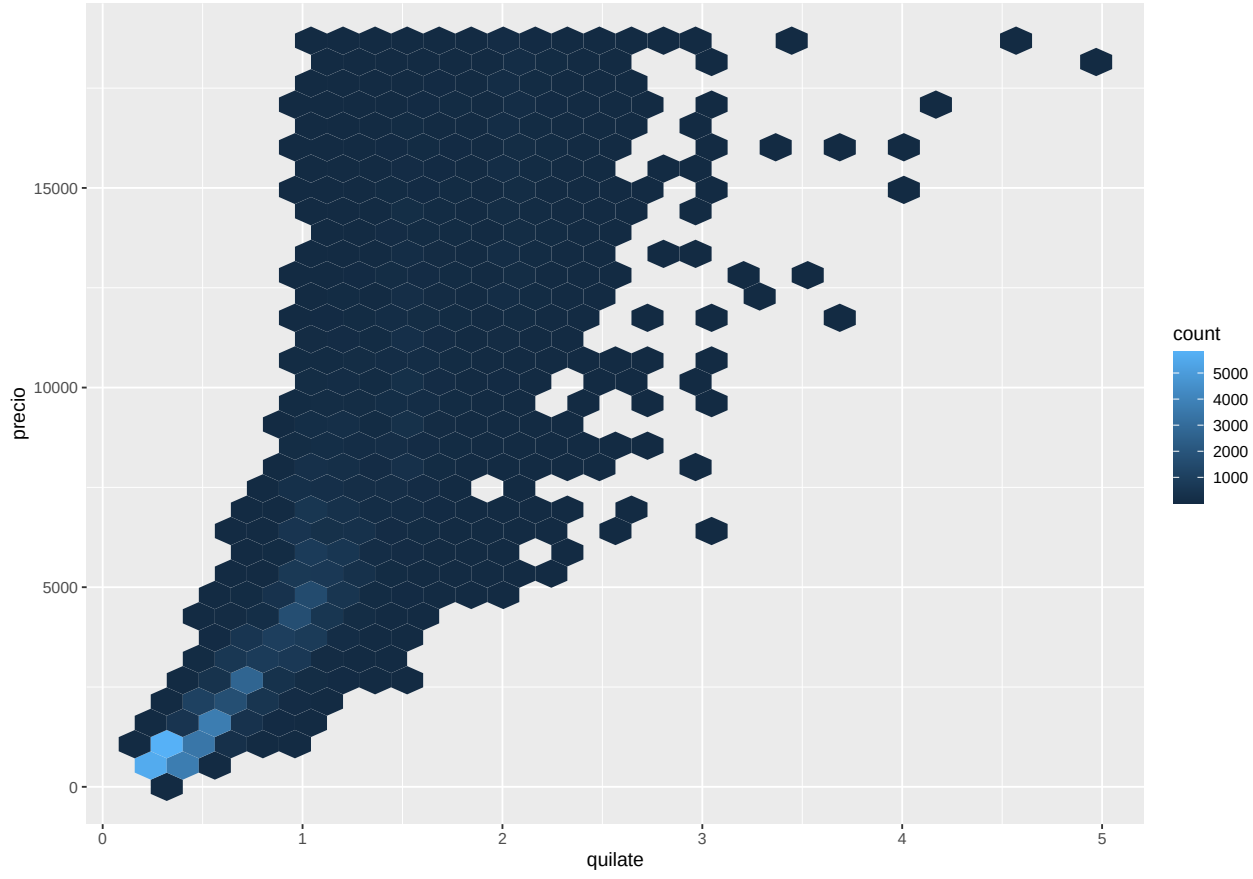


Diagrama de dispersión con contenedores hexagonales

```
# install.packages("hexbin")
library("hexbin")
ggplot(data = diamantes) +
  geom_hex(mapping = aes(x = quilate, y = precio))
```



Categorización de una variable continua

Otra opción es crear contenedores o intervalos con una de las variables continuas de manera de que pueda ser tratada como una variable categórica.

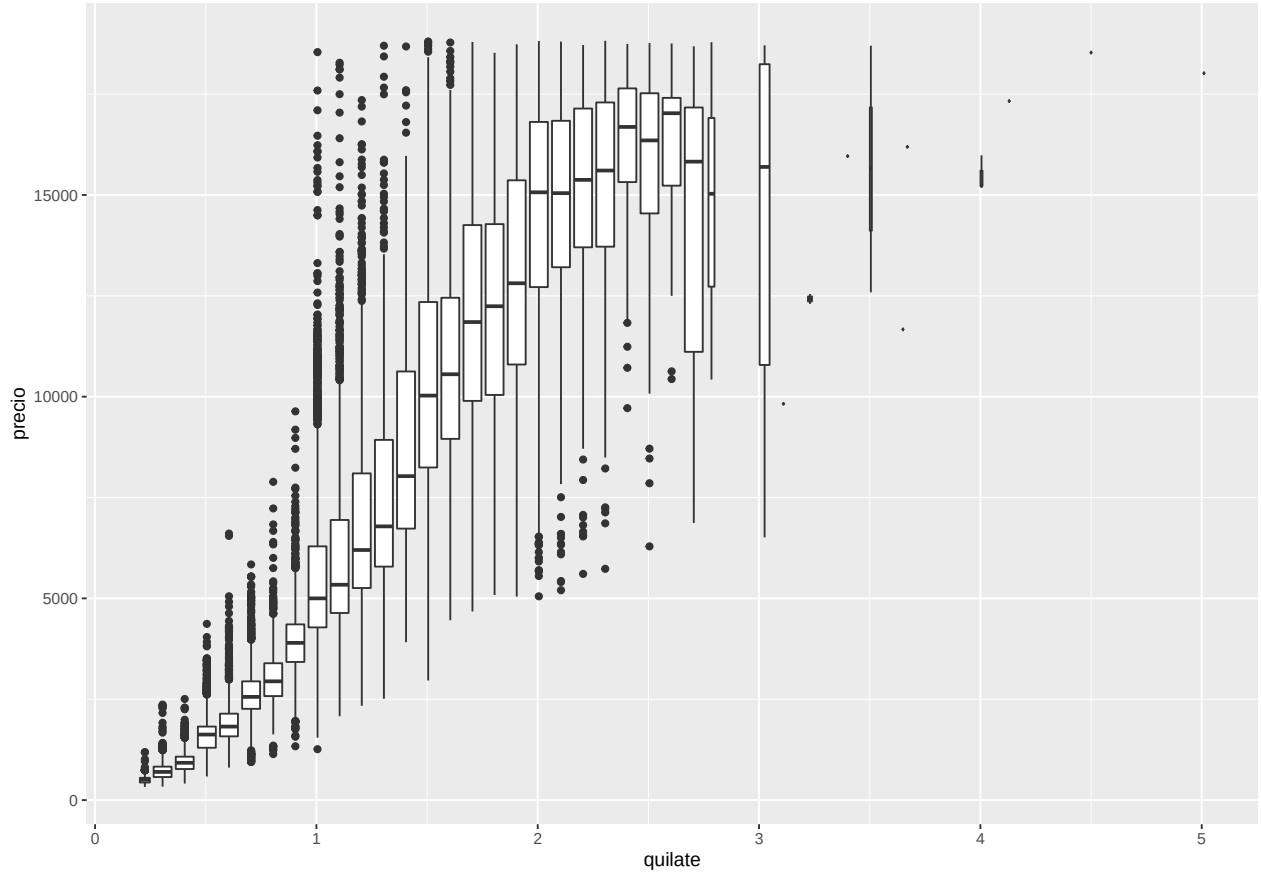
Luego, se puede utilizar alguna de las técnicas de visualización empleadas para representar la combinación de una variable categórica con una variable continua.

Por ejemplo, se puede segmentar la variable `quilate` en intervalos, para después graficar un diagrama de cajas para cada categoría o intervalo.

Con la función `cut_width` se pueden generar intervalos con una amplitud fija y una cantidad de datos variable en cada uno de ellos, mientras que con `cut_number` se obtienen intervalos de amplitud variable pero frecuencia fija (aproximadamente).

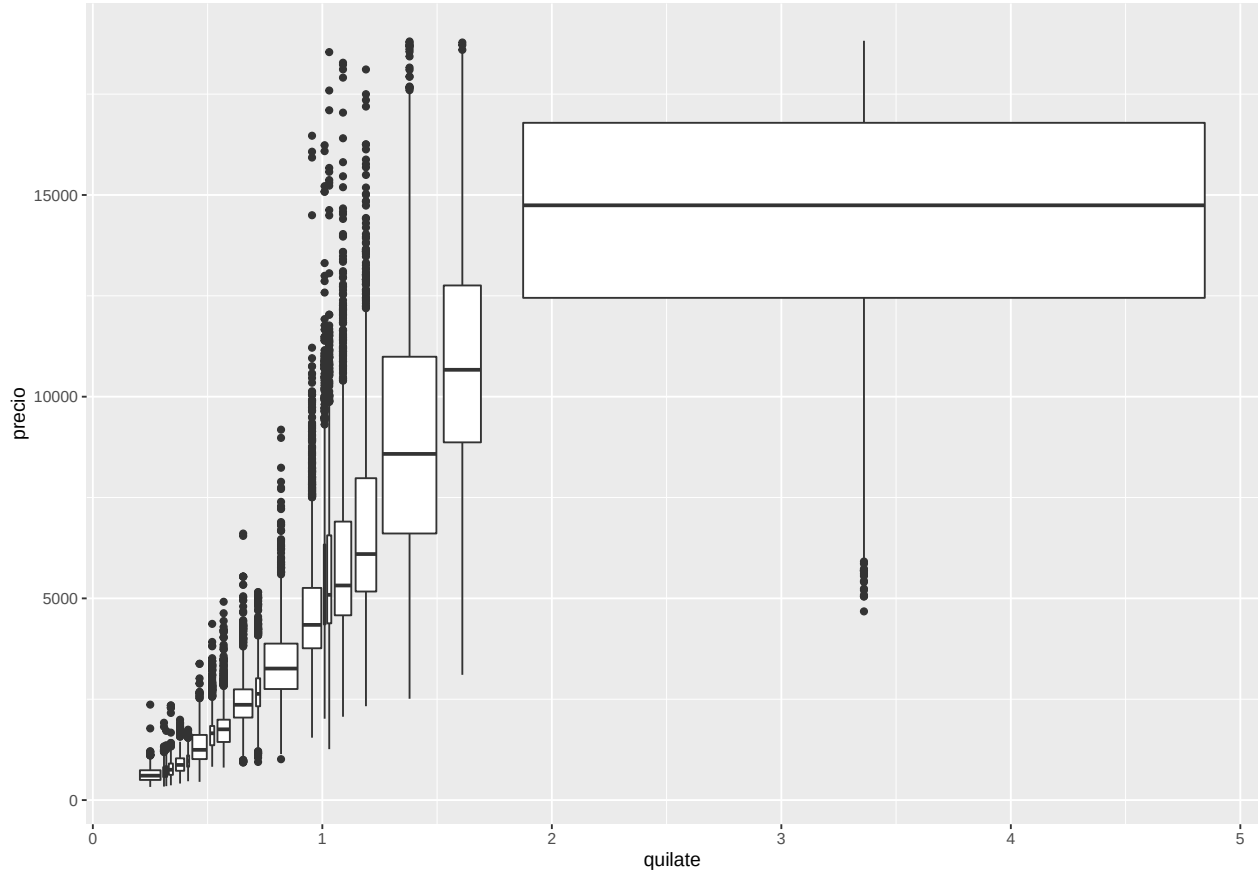
Categorización con intervalos de amplitud fija

```
ggplot(data = diamantes, mapping = aes(x = quilate, y = precio)) +
  geom_boxplot(mapping = aes(group = cut_width(quilate, 0.1)))
```



Categorización con intervalos de frecuencia fija

```
ggplot(data = diamantes, mapping = aes(x = quilate, y = precio)) +
  geom_boxplot(mapping = aes(group = cut_number(quilate, 20)))
```



Ejercicios

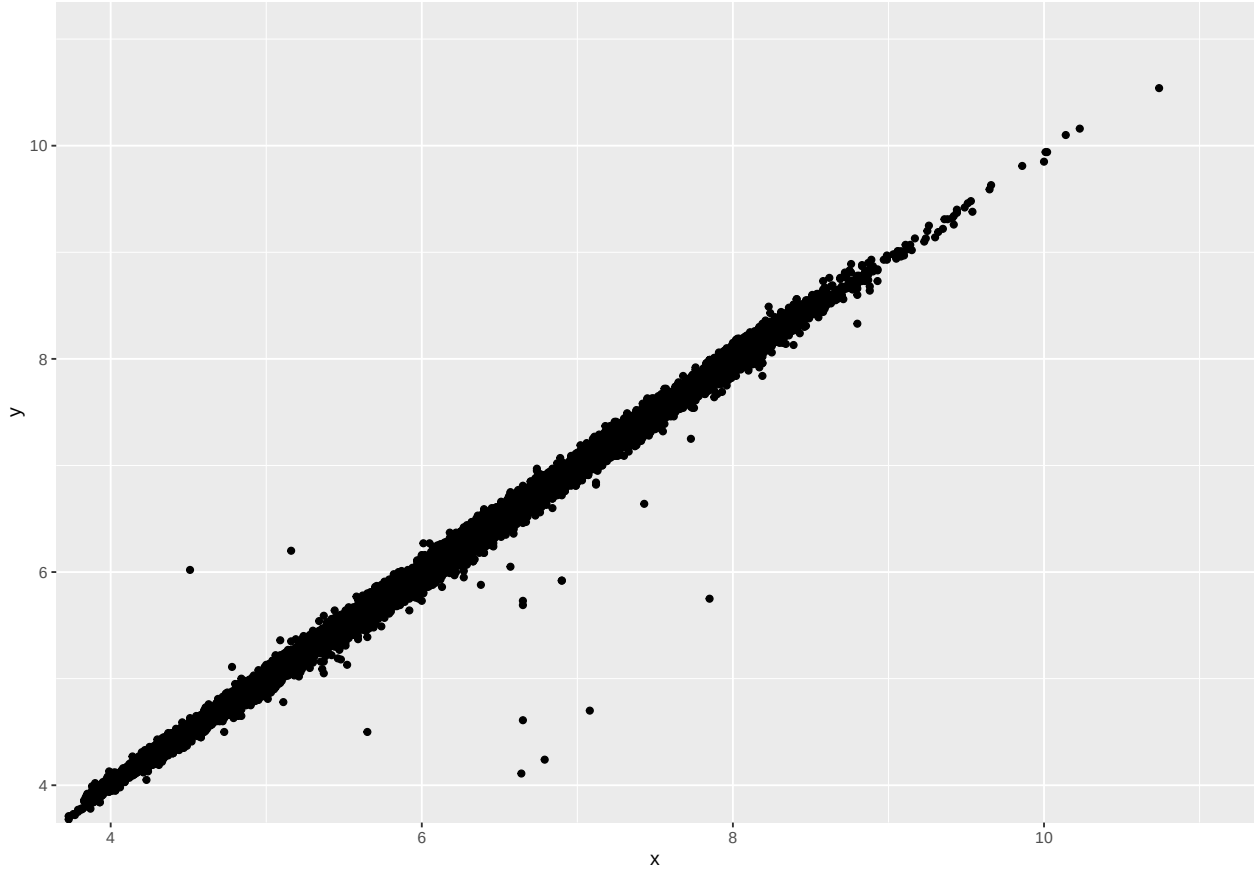
5. Los gráficos bidimensionales revelan observaciones atípicas que podrían no ser visibles en gráficos unidimensionales.

   Por ejemplo, algunos puntos en la gráfica a continuación tienen una combinación inusual de valores `x` e `y`, que hace que algunos puntos sean valores atípicos aún cuando sus valores x e y parecen normales cuando son examinados de manera individual.

   ¿Por qué es mejor usar un diagrama de dispersión que un diagrama basado en intervalos en este caso?

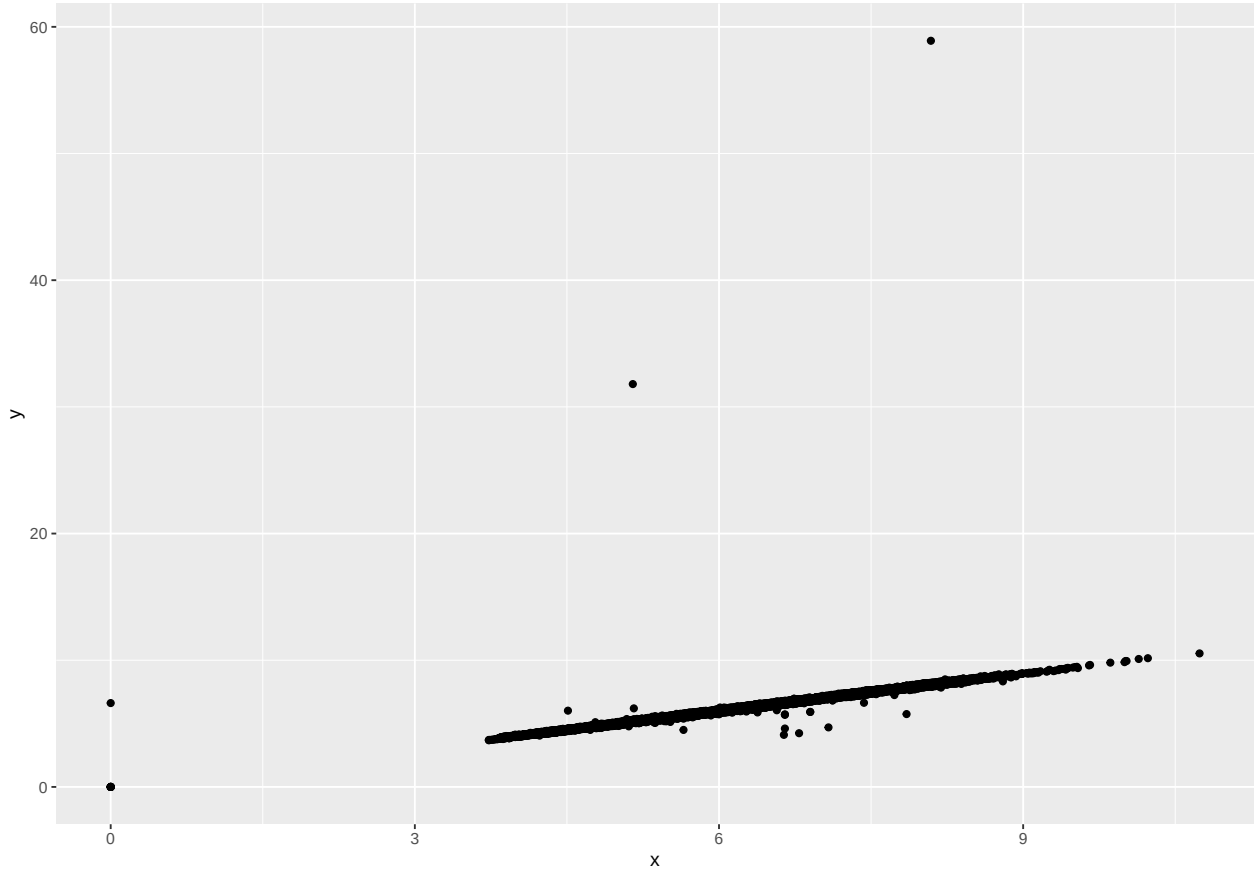
Ejercicios: diagrama de dispersión (dato del ejercicio)

```
ggplot(data = diamantes) +
  geom_point(mapping = aes(x = x, y = y)) +
  coord_cartesian(xlim = c(4, 11), ylim = c(4, 11))
```



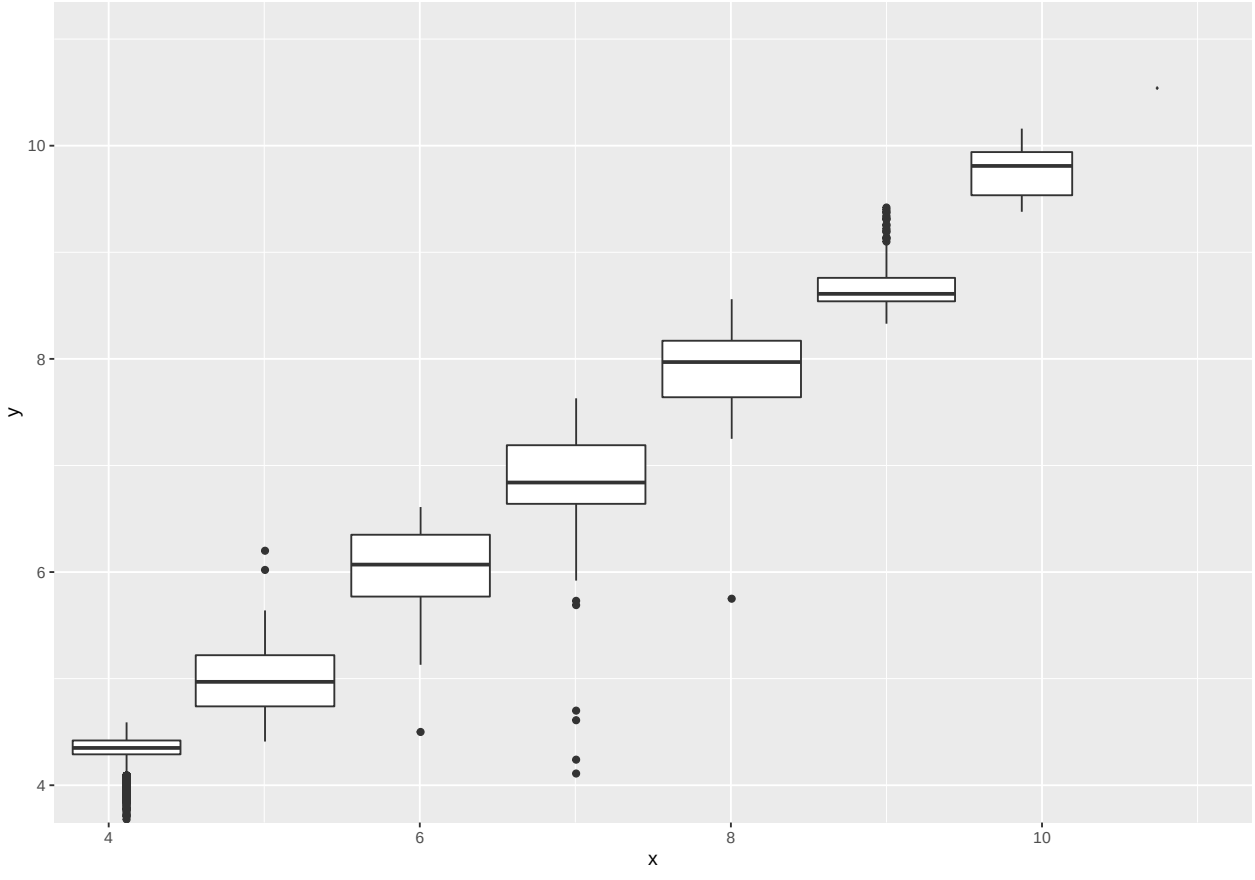
¿Por qué coord_cartesian(xlim = c(4, 11), ylim = c(4, 11)) ?

```
ggplot(data = diamantes) +
  geom_point(mapping = aes(x = x, y = y))
```



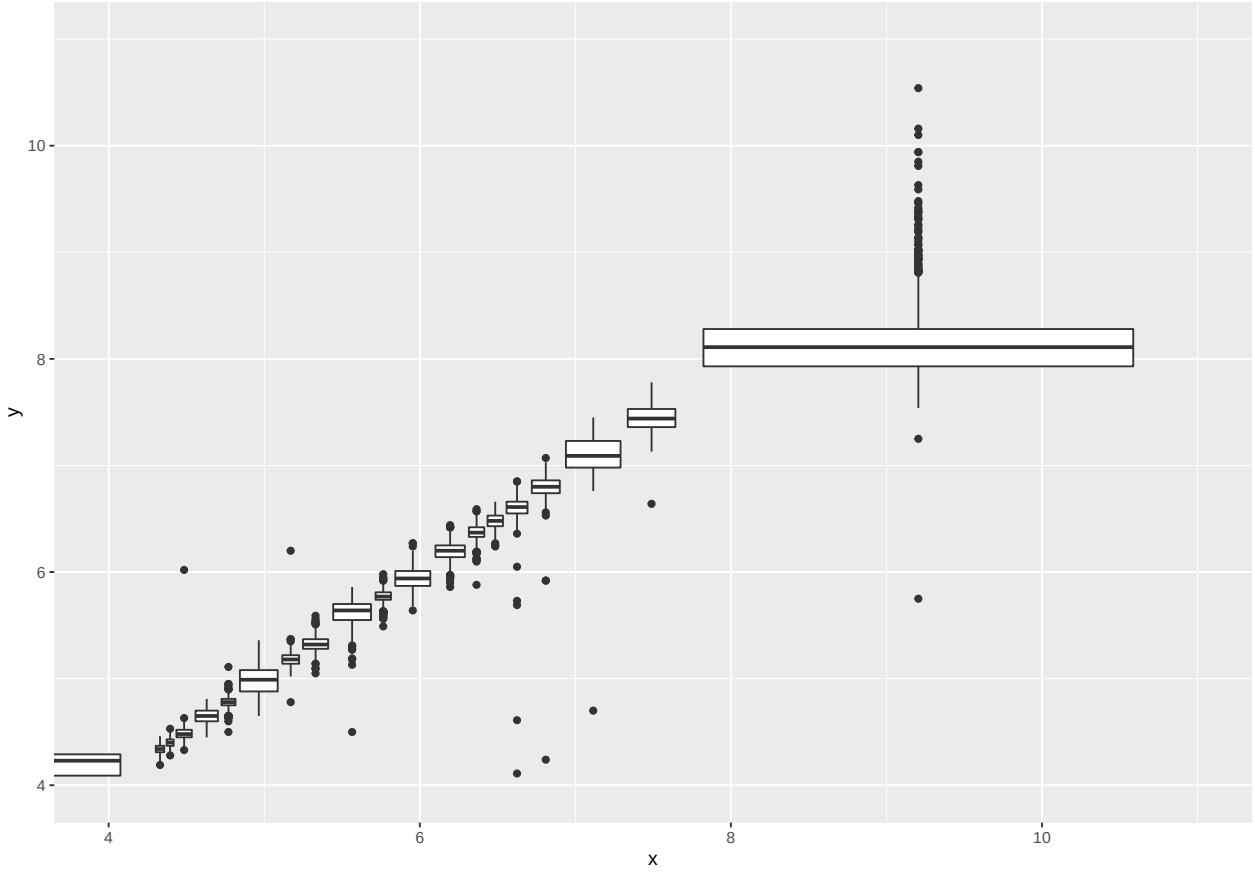
Ejercicios: diagrama basado en intervalos de amplitud fija

```
ggplot(data = diamantes, mapping = aes(x = x, y = y)) +
  geom_boxplot(mapping = aes(group = cut_width(x, 1))) +
  coord_cartesian(xlim = c(4, 11), ylim = c(4, 11))
```



Ejercicios: diagrama basado en intervalos de frecuencia fija

```
ggplot(data = diamantes, mapping = aes(x = x, y = y)) +
  geom_boxplot(mapping = aes(group = cut_number(x, 20))) +
  coord_cartesian(xlim = c(4, 11), ylim = c(4, 11))
```



Patrones y modelos

Los patrones en los datos dan pistas acerca de las relaciones entre variables.

Si existe una relación sistemática entre dos variables, esto aparecerá como un patrón en los datos (si hay suficientes).

Al encontrr un patrón, conviene hacerse las siguientes preguntas:

(i) ¿este patrón podría ser mera coincidencia (por azar)?;

(ii) ¿cómo podría describirse la relación sugerida por este patrón?;

(iii) ¿qué tan fuerte es la relación sugerida por este patrón?;

(iv) ¿qué otras variables podrían afectar la relación?;

(v) ¿cambia esta relación al examinar distintos subgrupos?[1]

[1] ¿Paradoja de Simpson?

Patrones y modelos

Los patrones son una de las herramientas más útiles para quienes hacen ciencia de datos.

Si dos variables varían de manera conjunta, entonces es posible utilizar los valores de una variable para hacer mejores predicciones sobre los valores de la otra.

Si la covariación es producto de una relación causal (un caso especial), entonces se puede usar el valor de una variable para controlar el valor de la segunda.

Los modelos son una herramienta para extraer patrones de los datos.

Patrones y modelos: un ejemplo

Por ejemplo, resulta difícil entender la relación entre `corte` y `precio` en los datos `diamantes`, porque `corte` y `quilate`, así como `quilate` y `precio`, están estrechamente relacionadas.

Es posible usar un modelo[2] para remover la fuerte relación entre `precio` y `quilate` de manera que podamos explorar otras relaciones existentes.

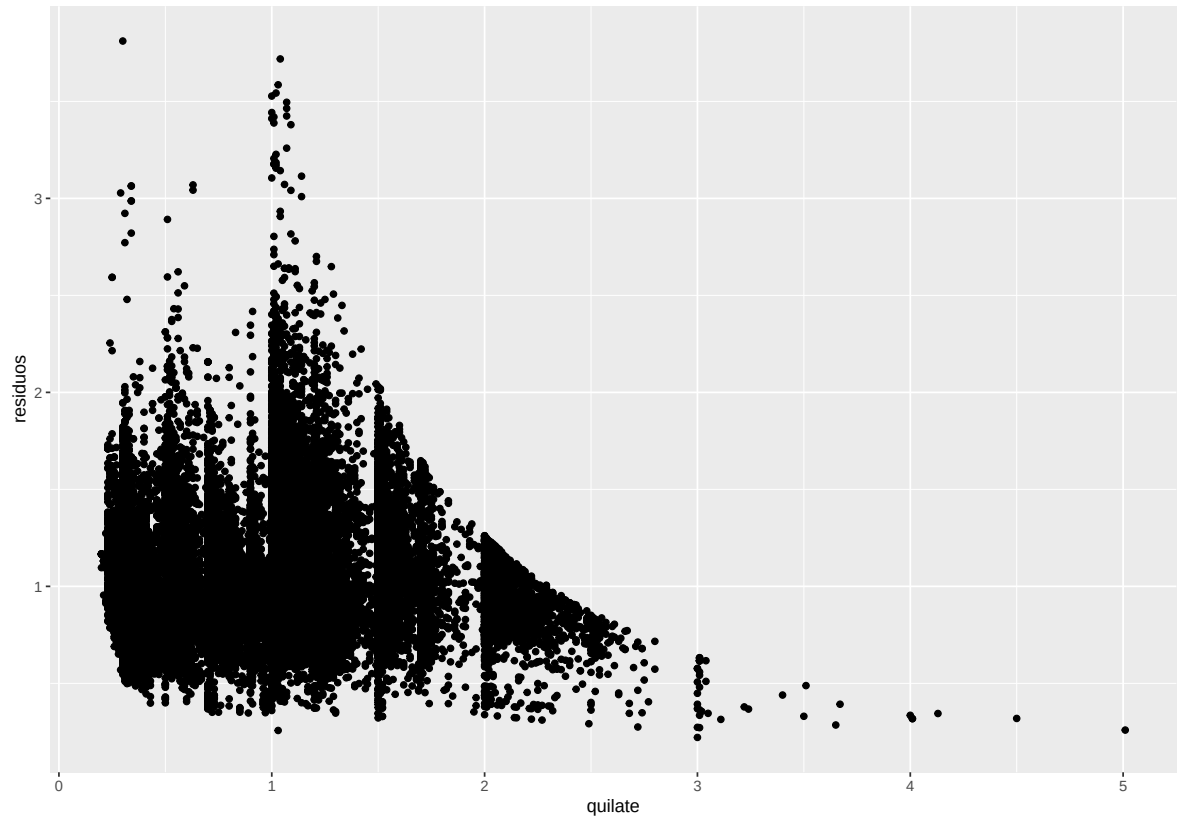
El siguiente modelo explica/predice `precio` a partir de `quilate` y después calcula los residuos (diferencia entre la variable predicha por el modelo y el valor real).

Los residuos nos informan acerca del precio de un diamante, una vez que el efecto que los quilates tienen sobre esta variable ha sido removido (la parte del `precio` no explicada por los `quilates`).

[2] Log lineal, posiblemente motivado por el patrón del diagrama de dispersión.

Residuos del modelo ajustado

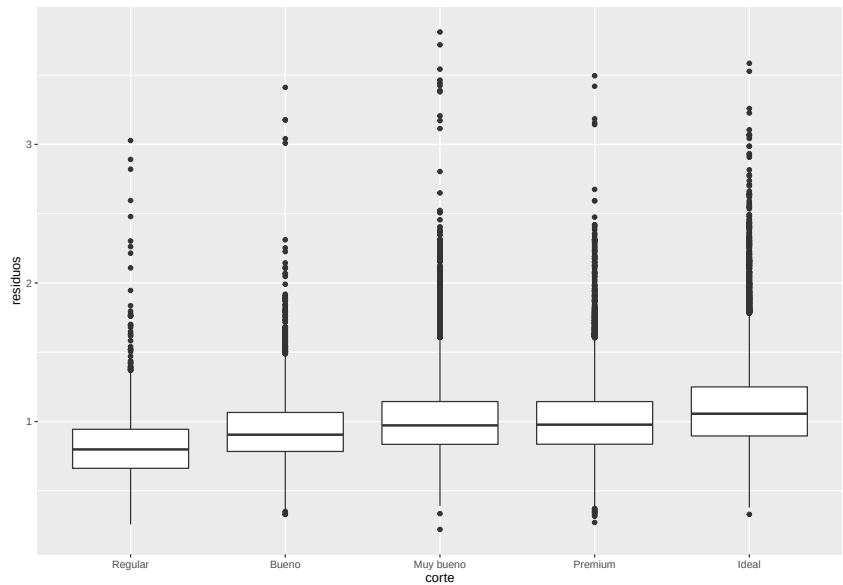
```
library(modelr) # paquete con "Modelling Functions that Work with the Pipe"
modelo <- lm(log(precio) ~ log(quilate), data = diamantes)
diamantes2 <- diamantes %>%
  add_residuals(modelo) %>% # agrega residuos al dataframe diamantes2
  mutate(residuos = exp(resid)) # transforma los residuos
ggplot(data = diamantes2) +
  geom_point(mapping = aes(x = quilate, y = residuos))
```



Efecto (residual) entre `corte` y `precio`

Una vez removida la fuerte relación entre `quilate` y `precio`, se puede observar lo esperado acerca de la relación entre `corte` y `precio`: los diamantes de mejor calidad son (en promedio) más costosos según su tamaño.

```
ggplot(data = diamantes2) +
  geom_boxplot(mapping = aes(x = corte, y = residuos))
```



Observación: más sobre modelado estadístico y `modelr` en la sección Modelos.

Argumentos en `ggplot2`

A medida que nos alejamos de estos capítulos introductorios, usaremos expresiones más concisas para escribir código.

Los primeros dos argumentos de `ggplot` son `data` y `mapping`, y los primeros dos argumentos de `aes` son `x` e `y`.

En lo que resta del libro no escribiremos esos nombres de argumentos.[3]

Esta es una cuestión importante en cuanto a programación se refiere, y hablaremos más sobre esto en el capítulo sobre funciones (sección Programar).

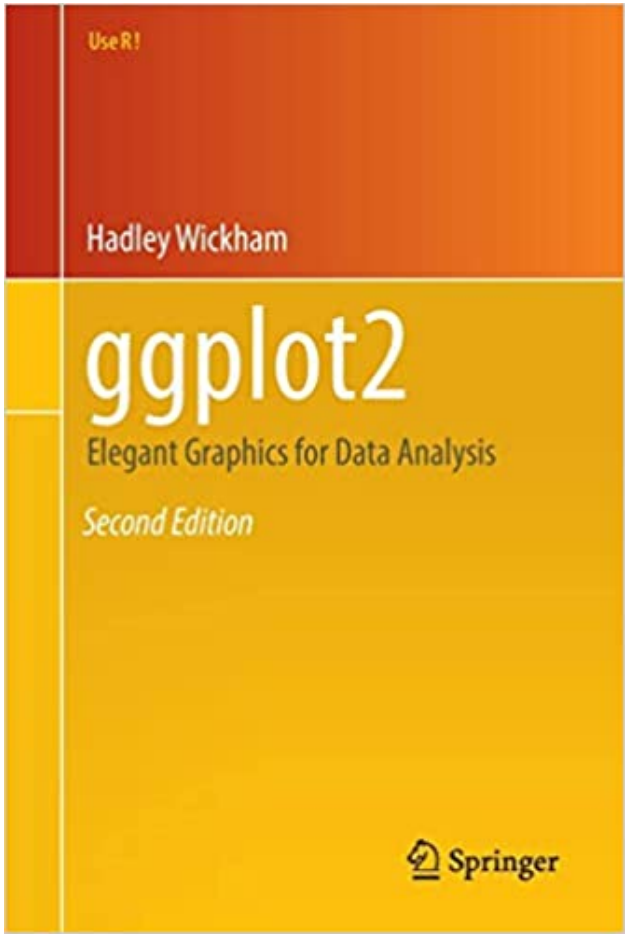
[3] Siempre que estén en el orden adecuado.

Ahorrando código: un ejemplo

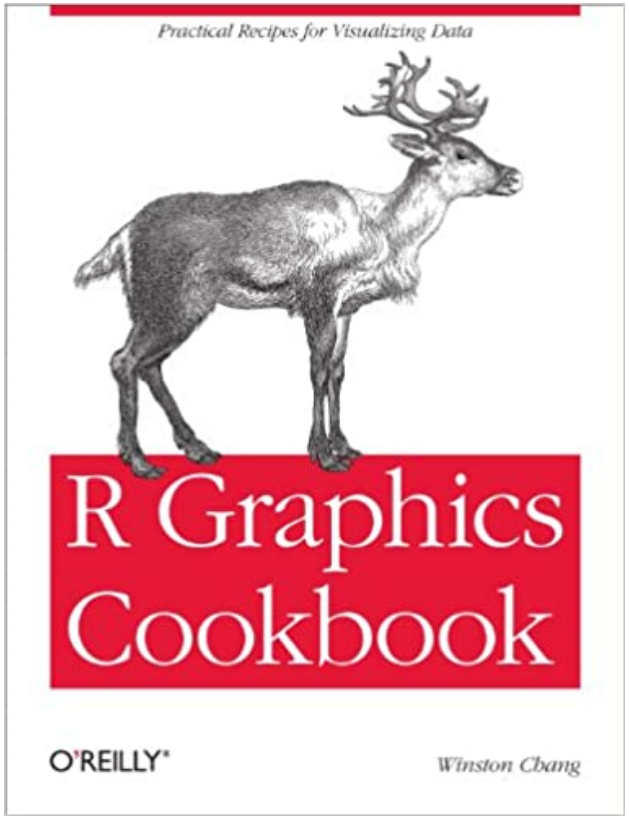
```
ggplot(data = diamantes2) +
  geom_boxplot(mapping = aes(x = corte, y = residuos))
```

```
ggplot(diamantes2) + geom_boxplot(aes(corte, residuos))
```

Aprendiendo más (un libro sugerido por el autor)

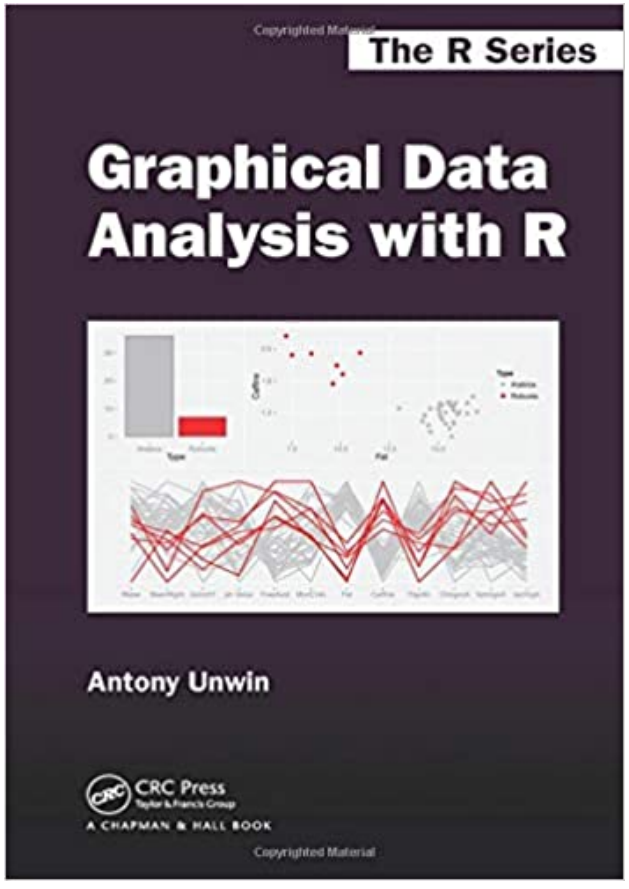


Aprendiendo más (otro libro sugerido por el autor)



Disponible en: http://http://www.cookbook-r.com/Graphs/.

Aprendiendo más (y otro libro sugerido por el autor)



Una última sugerencia (personal)

Un artículo que plantea una visión de la *Ciencia de Datos* trazando sus orígenes en el *Análisis Exploratorio de Datos* que planteaba John Tukey como reforma de la *Estadística* académica tradicional hace más de 50 años:

"50 years of Data Science", David Donoho, 2017

Disponible en:

https://www.tandfonline.com/doi/pdf/10.1080/10618600.2017.1384734?needAccess=true