Análisis Exploratorio de Datos con R

Eduardo Fernández

Basado en: Capítulo 7 de: R for Data Science, Wickham& Grolemund, 2017

ggplot2 cheat sheet ggplot2 cheat sheet en español

Comienzo en AED en R

- Instalar ggplot y tidyverse

- Ponerse como objetivo comprender los datos
  - Generar preguntas como herramienta para comprender los datos.
  - Eso ayuda a focalizarse y elegir qué gráficos, modelos y transformaciones hacer.
  - Es un proceso creativo.
  - Generar muchas preguntas para generar preguntas buenas.

  - Dos tipos claves de preguntas:
    - ¿Qué tipo de variación/distribución ocurre **dentro de cada variable**?
    - ¿Qué tipo de relación/covariación ocurre **entre las variables**?

Conceptos Iniciales

- **Variable:** cantidad, cualidad o propiedad que se mide de algo.
- **Valor:** estado de la variable cuando se mide.
- **Observación:** conjunto de medidas realizada en condiciones similares (p.ej: todas las observaciones son realizadas al mismo tiempo sobre un objeto)
- **Datos Tabulares:** conjunto de valores, donde cada valor está asociado a una variable y a una observación.

Los mismos conceptos de [Wickham 2014]

Variación de las variables

- Tendencia natural de los valores de una variable a cambiar entre medida y medida.
- Cada variable tiene su propio patrón de variación, que revela información interesante. La mejor forma de comprender el patrón es visualizar la distribución de los valores de la variable.

Variación de las variables

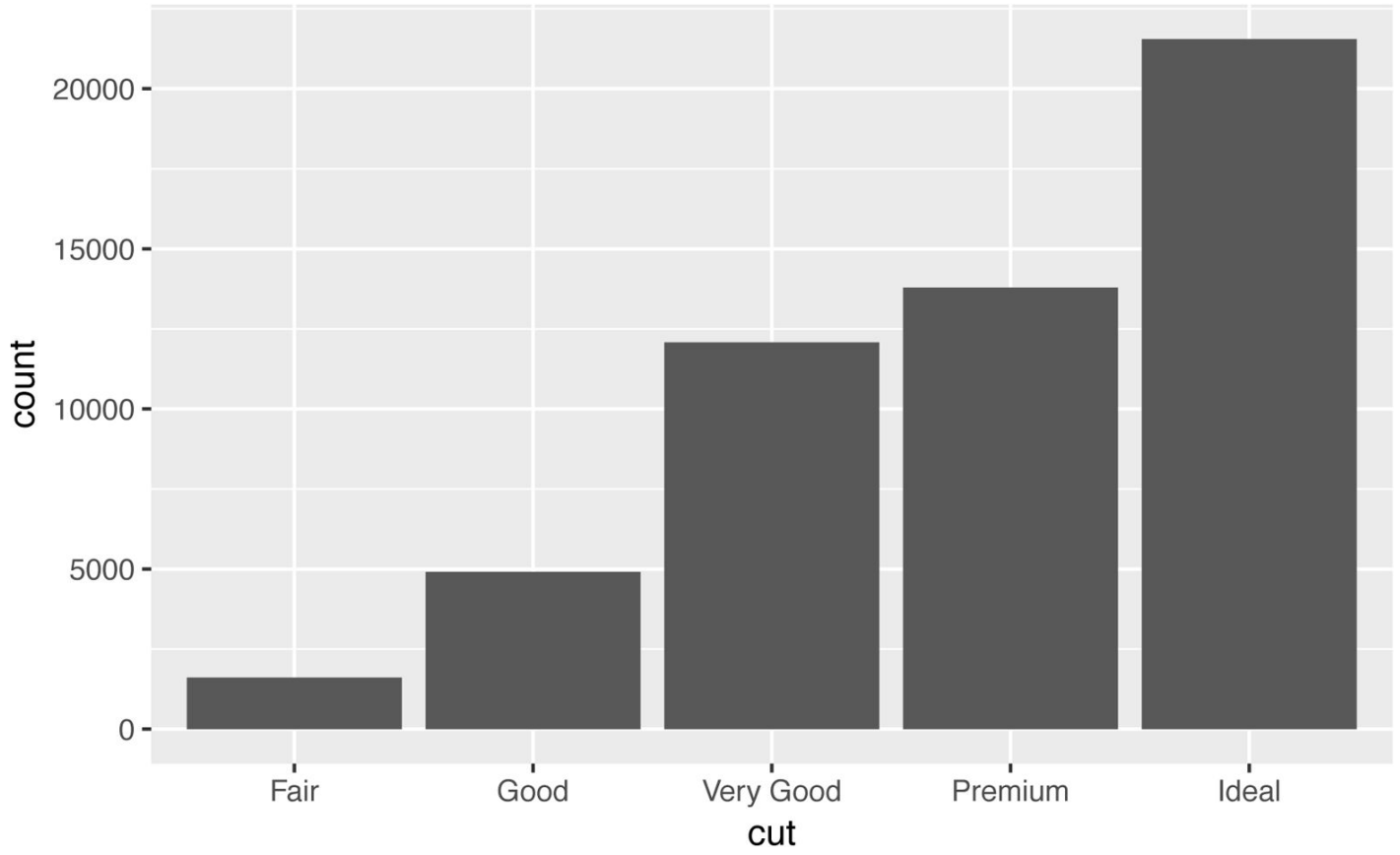
#Visualización de la distribución de los cortes de diamante.

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut))
```

#lo mismo se puede realizar manualmente

```
diamonds %>%count(cut)
```

```
# A tibble: 5 x 2
  cut         n
  <ord>     <int>
1 Fair       1610
2 Good       4906
3 Very Good 12082
4 Premium   13791
5 Ideal     21551
```



Variación de las variables

Cuando la variable es continua puede convenir un histograma para ver la distribución de los datos.

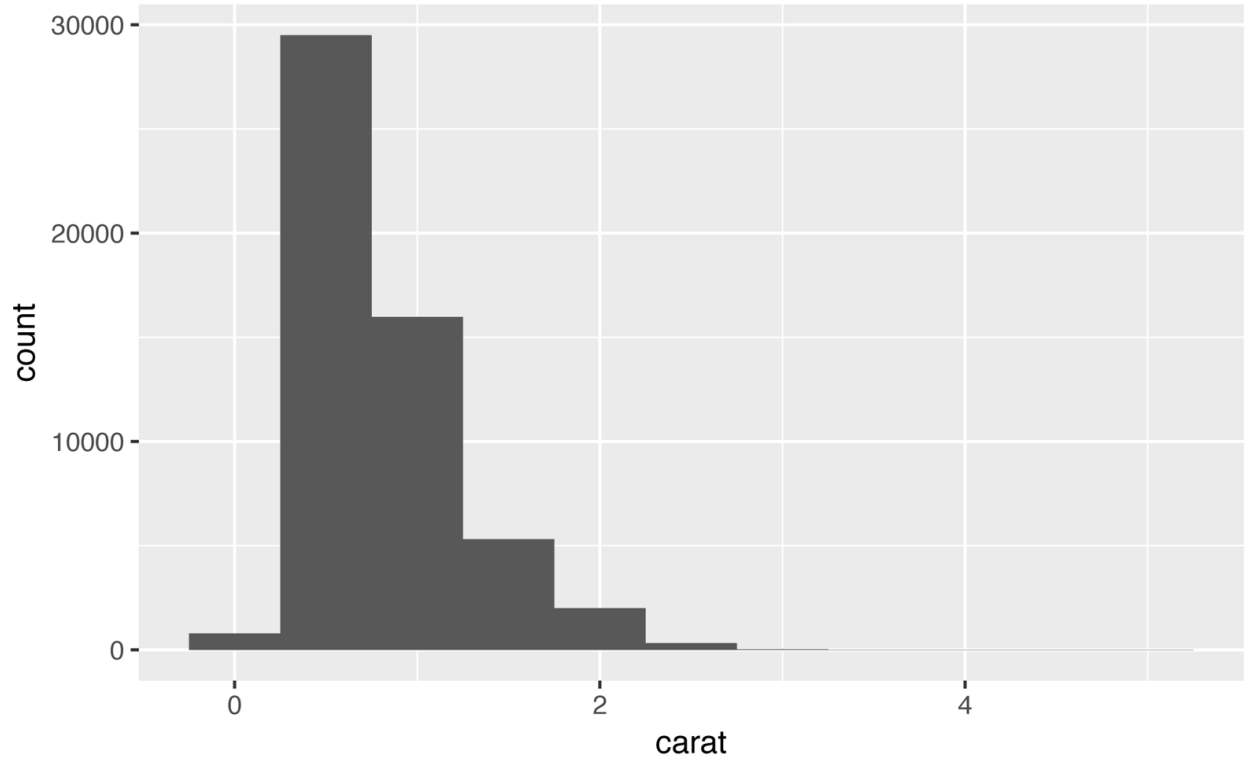
```
ggplot(data = diamonds) +
  geom_histogram(mapping = aes(x = carat), binwidth = 0.5)
```

#También combinando count() con cut_width() …

```
diamonds %>% count(cut_width(carat, .5))
```

```
# A tibble: 11 x 2
   `cut_width(carat, 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
```

Estos valores “mueven” el plot a la izquierda.



Observando mejor la distribución

#smaller se queda con los diamantes con carat menores a 3

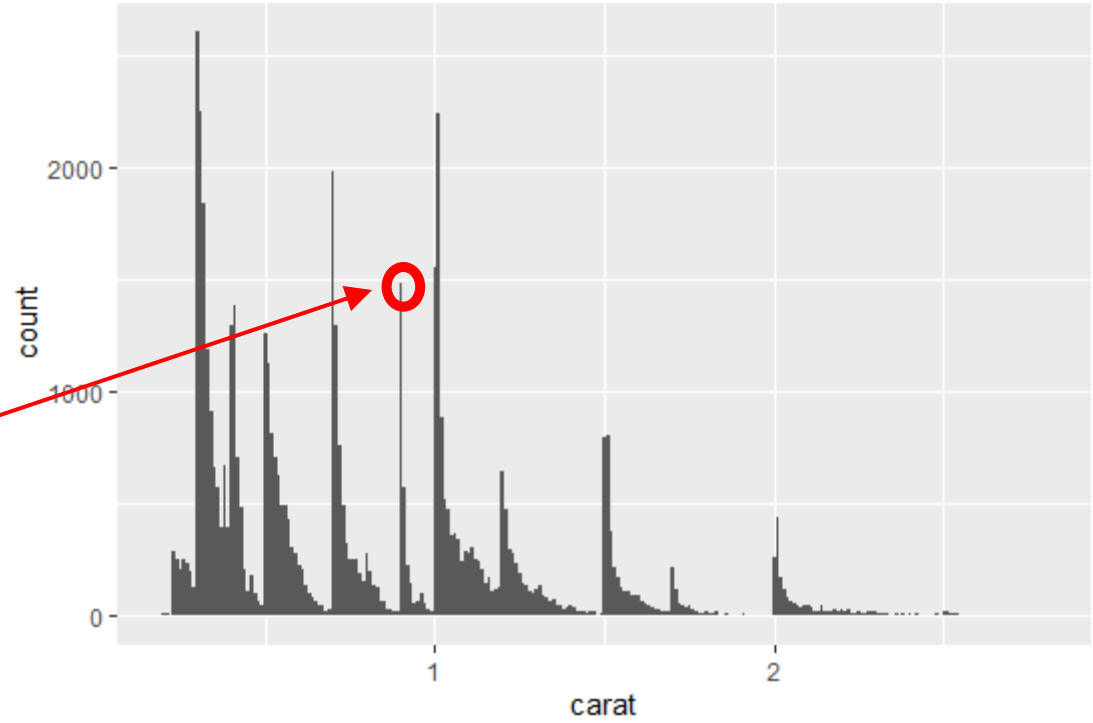
```
smaller <- diamonds %>% filter(carat < 3)
```

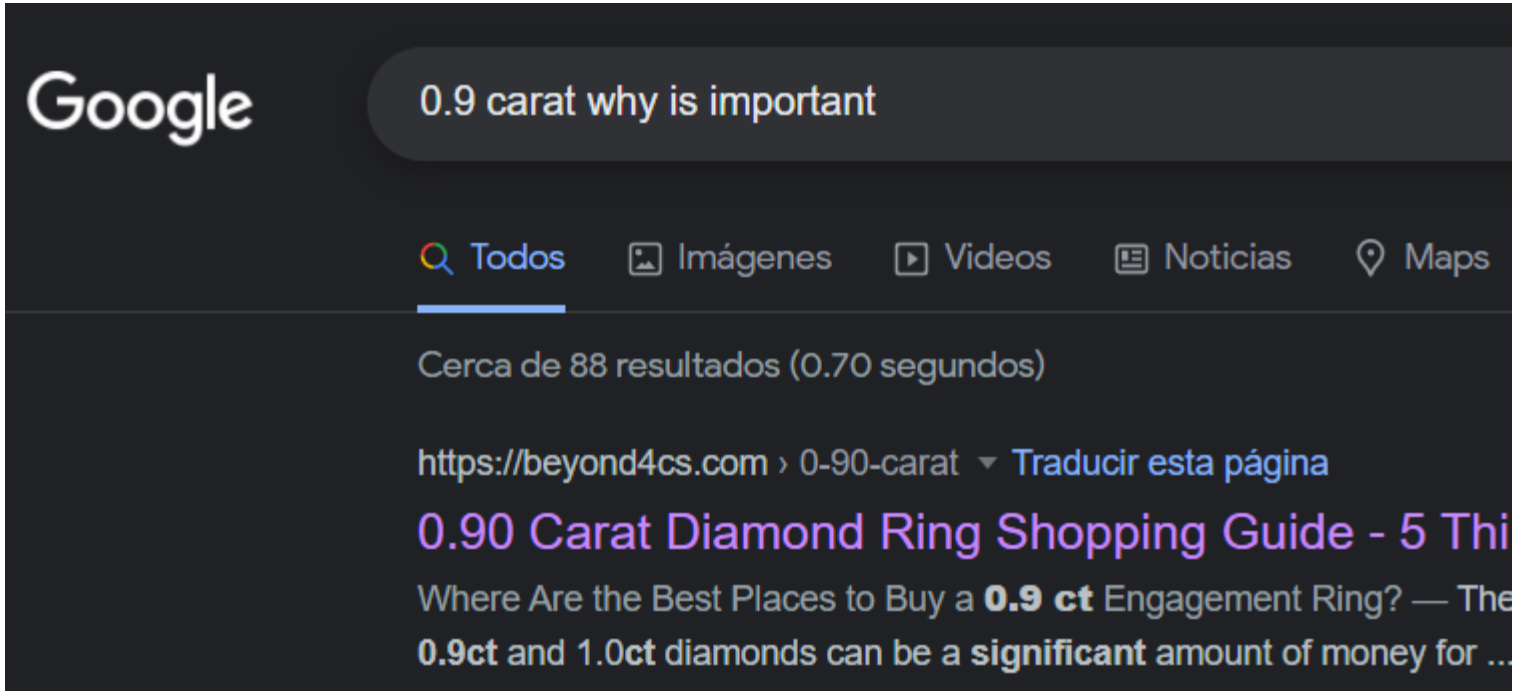
#Vemos a esos diamantes con mayor detalle.

```
ggplot(data = smaller, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.01)
```

¿Por qué hay acumulaciones en ciertos valores?

¿Por qué la concentración en 0.9 carat?





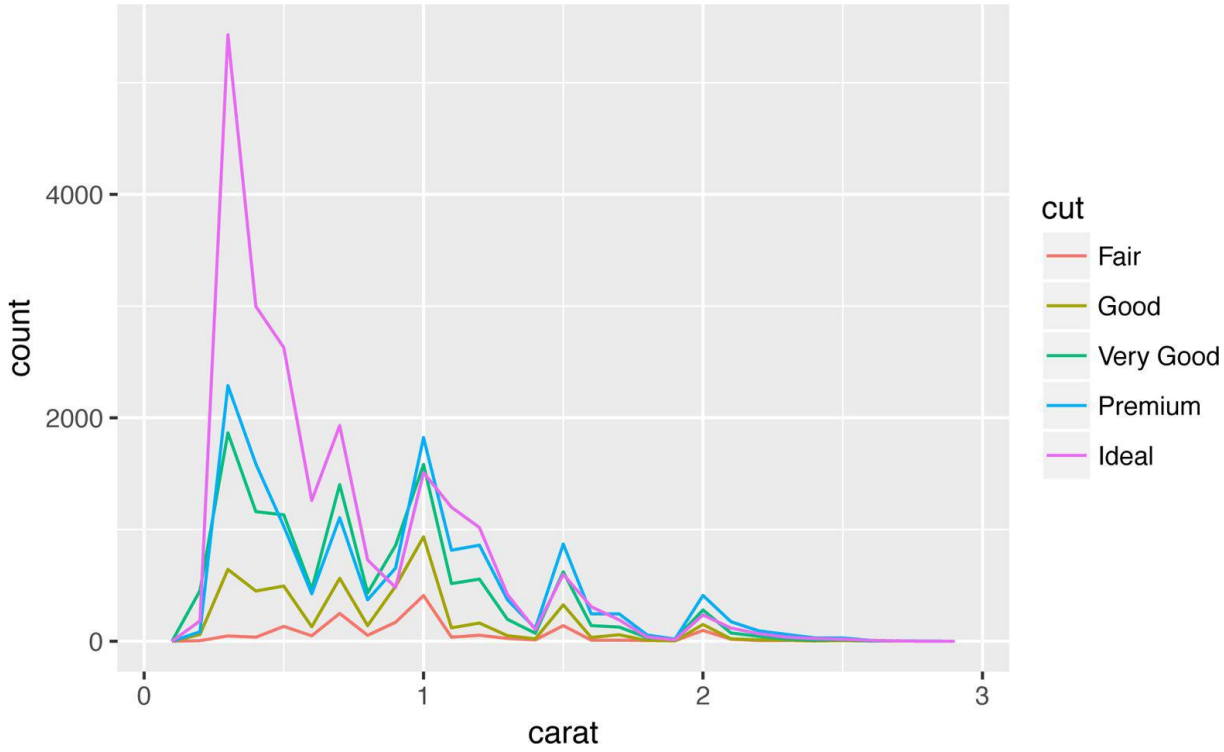
When it comes to buying a diamond engagement ring, 1 carat is the “magic” weight that most women desire. Due to its popularity, the prices of one carat diamonds are elevated because of supply and demand mechanics.

If you are on a tighter budget, an alternative is to buy a 0.90 carat diamond which is just below the psychological 1 carat mark. This way, you get a diamond ring which looks very similar in size to a 1 ct ring but avoid paying the price premium.

Superposición de histogramas: geom_freqpoly()

#Para realizar múltiples histogramas en el mismo plot

```
ggplot(data = smaller, mapping = aes(x = carat, color = cut)) +
  geom_freqpoly(binwidth = 0.1) +
  scale_color_brewer(palette = "Set1")
```



Preguntas que nos podemos hacer a partir de estos datos:

Preguntas comunes al analizar datos y gráficos:

- ¿Cuáles son los valores comunes/raros? ¿Por qué? ¿Es lo que se esperaba?
- ¿Puede ver patrones inusuales? ¿Cómo se pueden explicar?

Como un ejemplo, los histogramas anteriores sugieren las siguientes preguntas:

- ¿Por qué hay más diamantes en quilates enteros y fracciones de quilate más comunes?
- ¿Por qué hay más diamantes hacia la derecha de cada pico que hacia la izquierda?
- ¿Por qué no hay diamantes más grandes de 3 quilates?

Preguntas que nos podemos hacer a partir de estos datos:

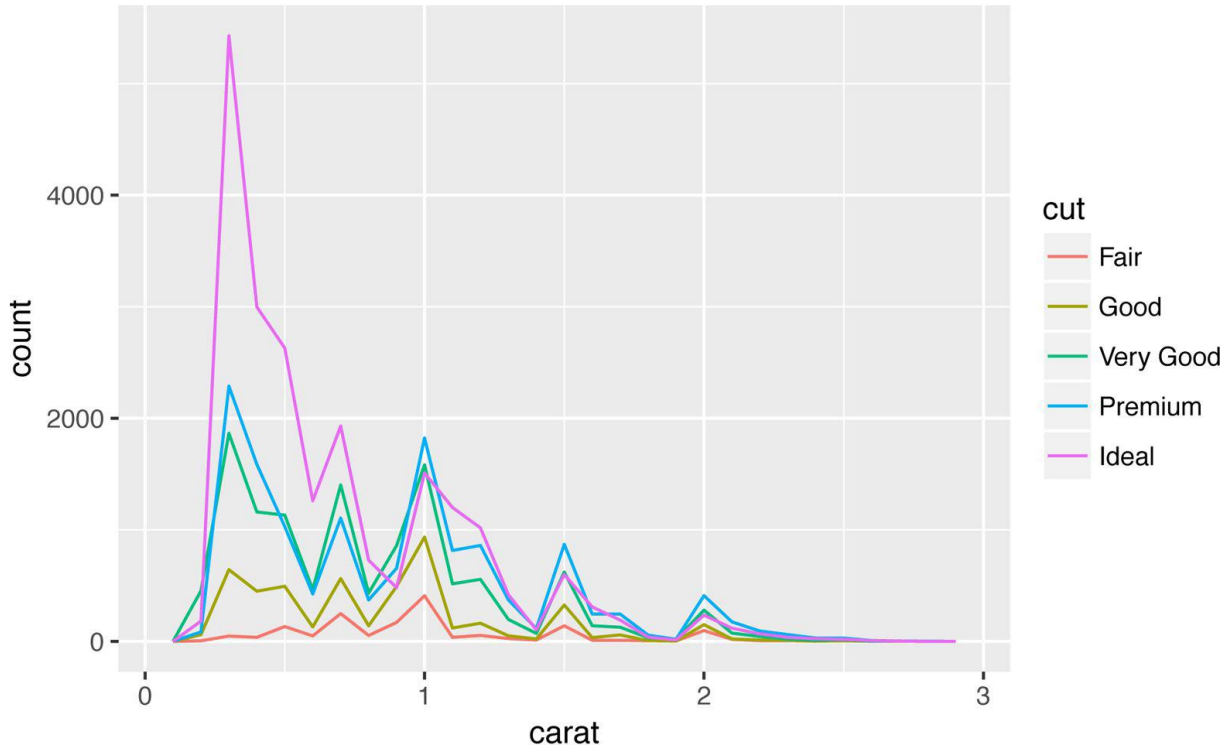
En general, los grupos de valores similares sugieren que existen subgrupos en sus datos.

Para comprender los subgrupos, pregúntese:

- ¿En qué características comunes tienen las observaciones dentro de cada grupo?

- ¿En qué se diferencian entre sí las observaciones en distintos grupos?

- ¿Cómo puede explicar o describir los grupos?

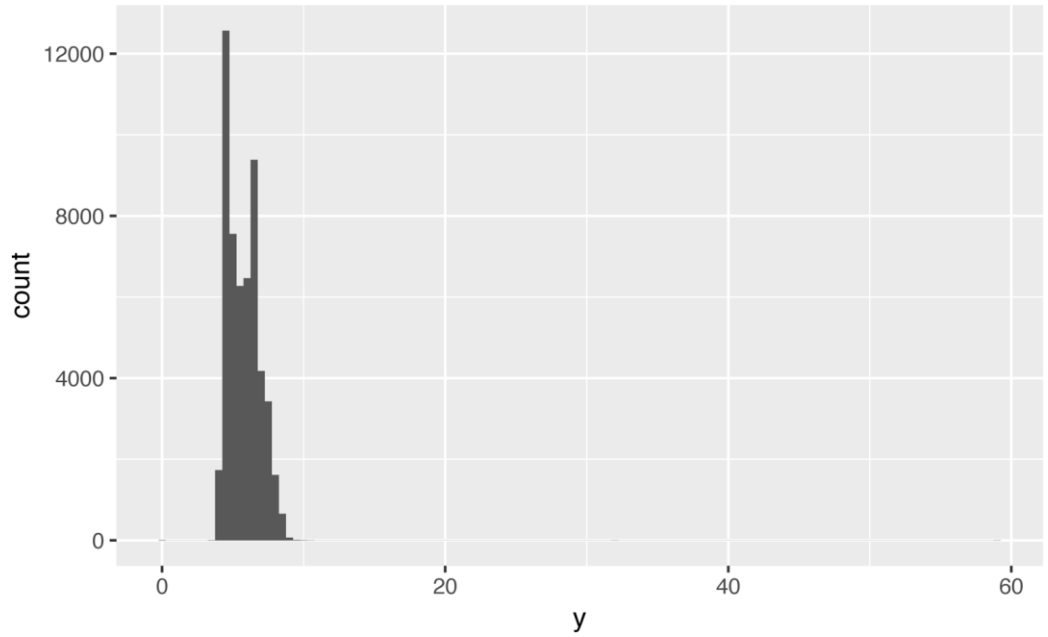


Valores Inusuales (outliers):

Outlier: observación distante del resto de los datos. Una observación que no sigue el patron usual de los datos. Puede deberse a un error en los datos, o puede ser por algo nuevo desconocido.

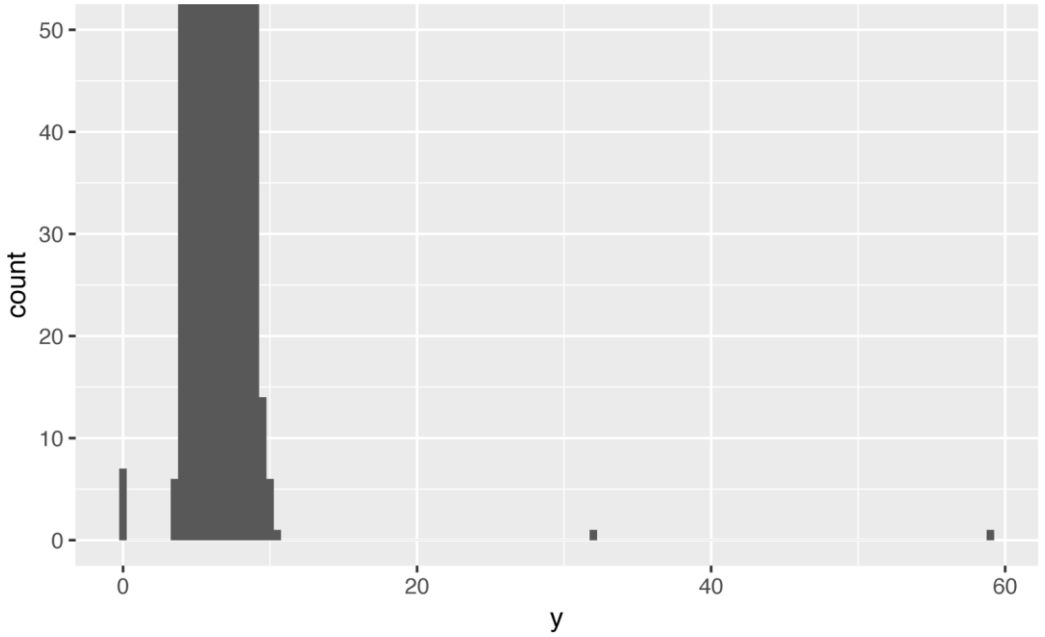
- **¿Por qué quedó contra la izquierda?**

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



- **coord_cartesian() para truncar la gráfica:**

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



Valores Inusuales

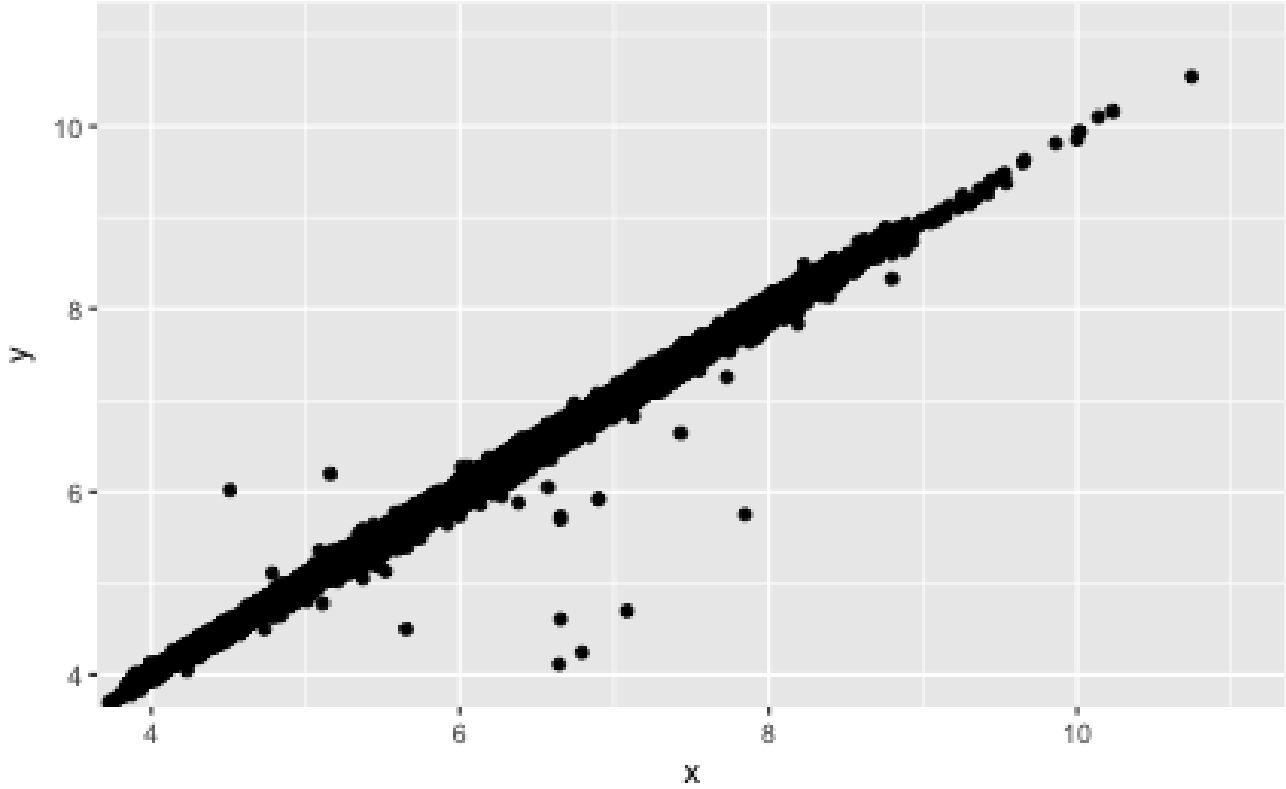
En general, ¿cómo tratar a los outliers?:

- Repetir los análisis con/sin outliers.
- Si tienen mínimo efecto en los resultados y no se puede deducir por qué están, convendría reemplazarlos por NA.
- Si tienen un efecto sustancial en los resultados, no deberían eliminarse sin la debida justificación. Deberá averiguar qué los causó, y en caso que se eliminen, aclararlo explícitamente.

Outliers por combinación de 2 variables

#Los outliers pueden existir en las combinaciones de dos variables (combinaciones poco probables de dos variables)

```
ggplot(data = diamonds2) +
    geom_point(mapping = aes(x = x, y = y)) +
     coord_cartesian(xlim = c(4, 11), ylim = c(4, 11))
```



¿Qué se hace con los outliers?

- **En el caso de diamonds …**

#O los eliminamos toda la observación (toda la fila) …

diamonds2 <- diamonds %>% **filter(between(y, 3, 20))**

#O borramos sólo los datos problemáticos pero dejamos el resto de la observación …

diamonds2 <- diamonds %>% **mutate(y = ifelse(y < 3 | y > 20, NA, y))** #analizar ifelse

Ojo con **NA**: agregar **na.rm = TRUE** en los cálculos

```
#ggplot2 pone Warnings cuando se eliman observaciones por falta de datos
>  ggplot(data = diamonds2, mapping = aes(x = x, y = y)) +
     geom_point()

  # na.rm = TRUE hace que no aparezcan los Warnings
>  ggplot(data = diamonds2, mapping = aes(x = x, y = y)) +   #los Warning desaparecen
     geom_point(na.rm = TRUE)

> mean(diamonds$y)
[1] 5.734526

#la media con datos faltantes hace que el resultado sea NA
> mean(diamonds2$y)
[1] NA

#al agregar na.rm = TRUE esos datos no son considerados
> mean(diamonds2$y,na.rm=TRUE)
[1] 5.733801
```

Covariación: relación entre variables

Veremos cómo visualizar relaciones entre:

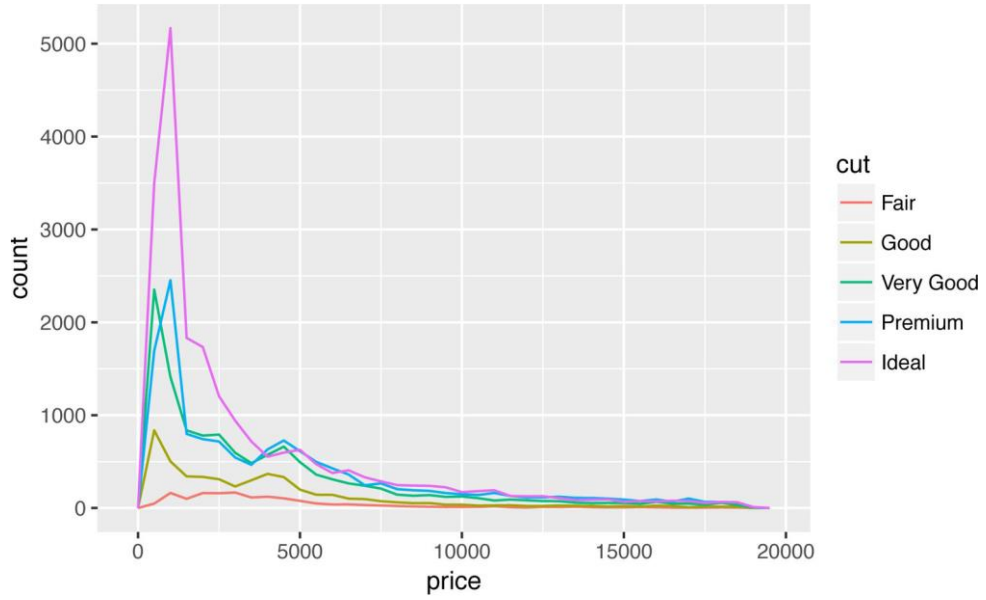
- Una variable categórica (nominal, ¿ordinal?) y una contínua (intervalo y ratio)
- Dos variables categóricas
- Dos variables contínuas

Covariación: categórica con continua

Analizar la variable continua según cada categoría de la otra variable.

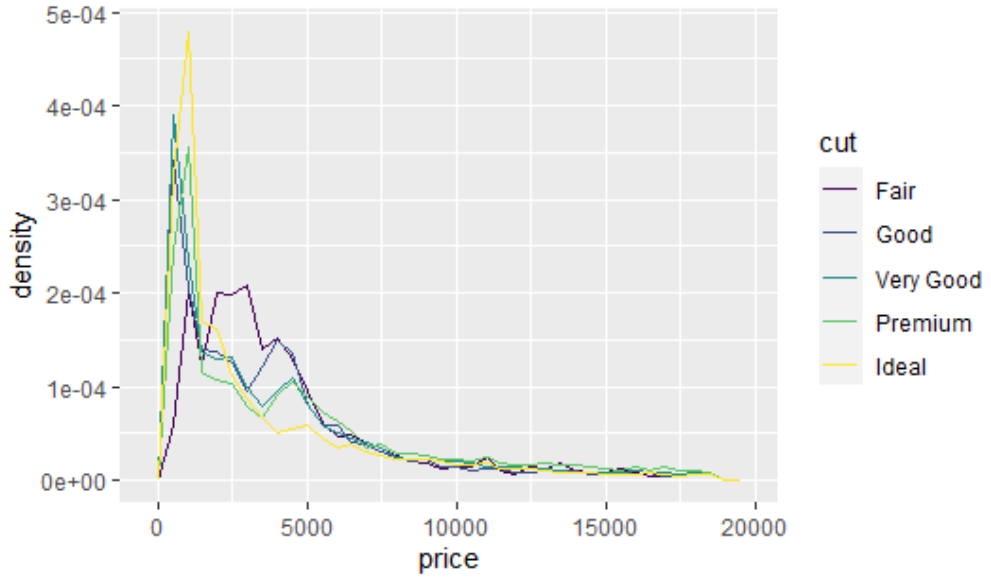
#Dificil comparar las distribuciones porque hay mucho de ideal y poco de Fair

```
ggplot(data = diamonds, mapping = aes(x = price)) +
  geom_freqpoly(mapping = aes(color = cut),
  binwidth = 500) #histograma por cada cut.
```



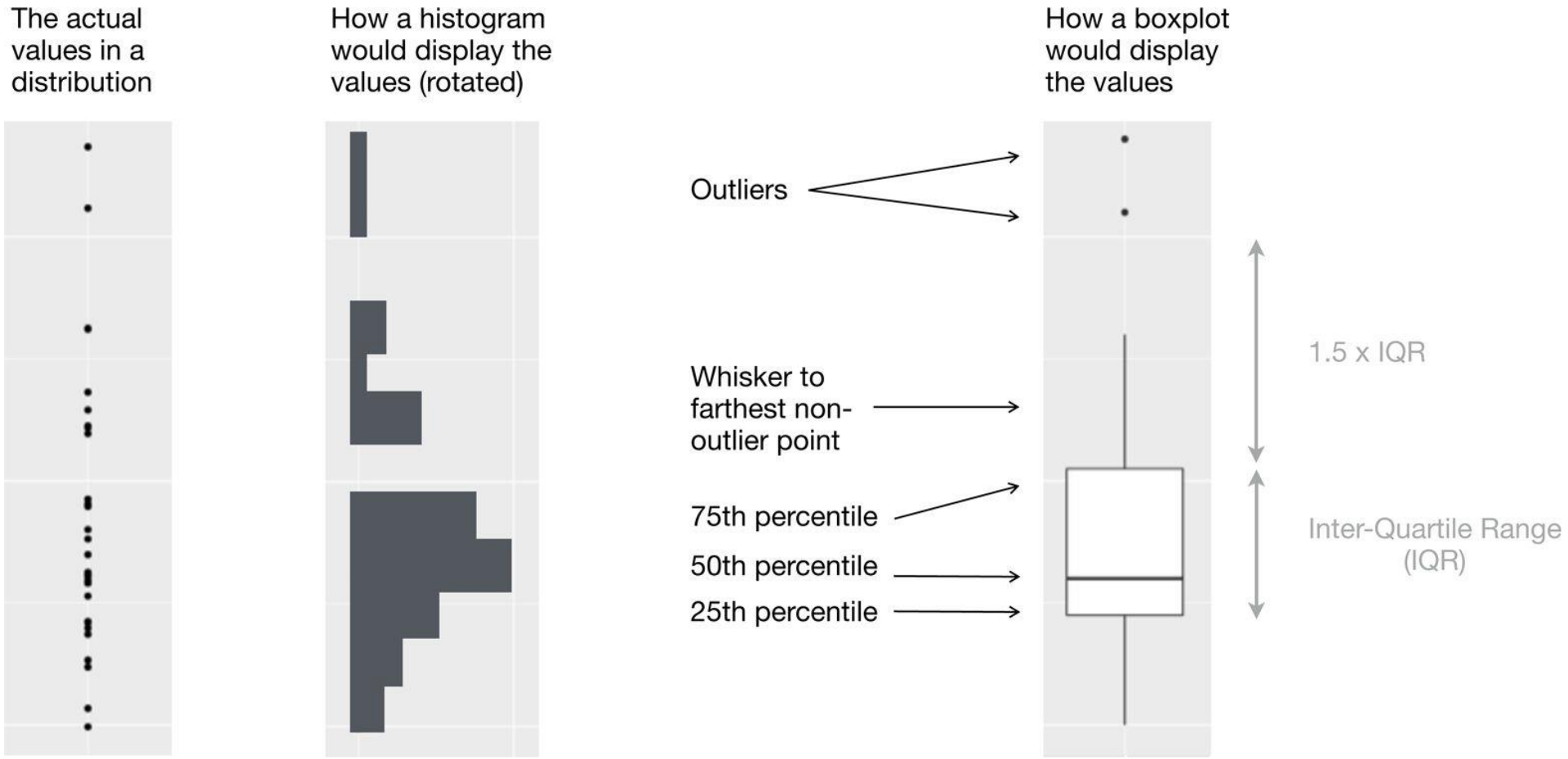
#y = ..density.. hace que el área bajo cada curva mida 1

```
ggplot(data = diamonds,mapping = aes(x = price,
y = ..density..)) +
geom_freqpoly(mapping = aes(color = cut), binwidth =500)
```



Covariación: categórica con continua

Otra opción para observar la distribución es utilizar un boxplot.

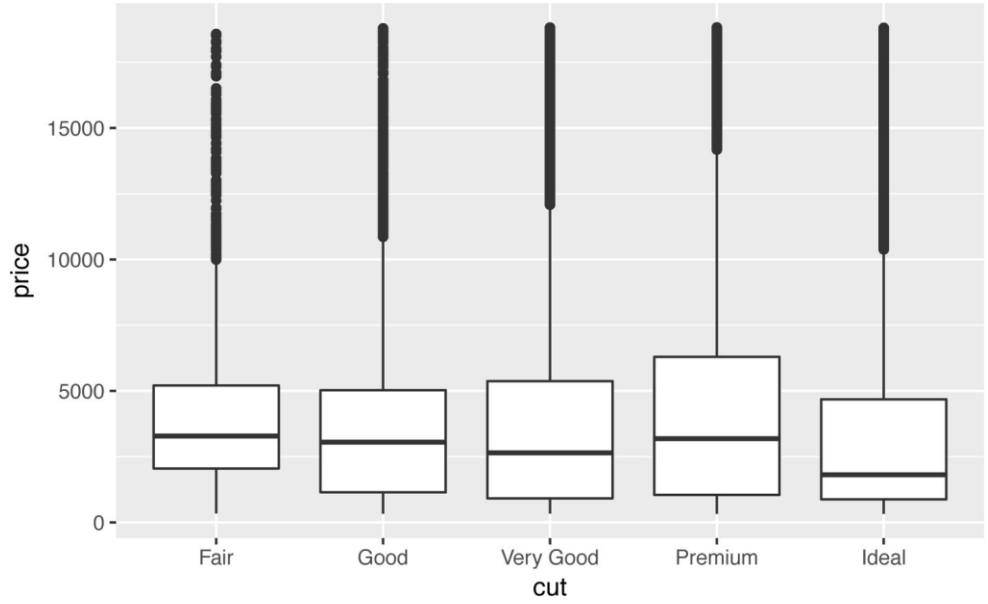


Covariación: categórica con continua

Otra opción para observar la distribución es utilizar **geom_boxplot()**

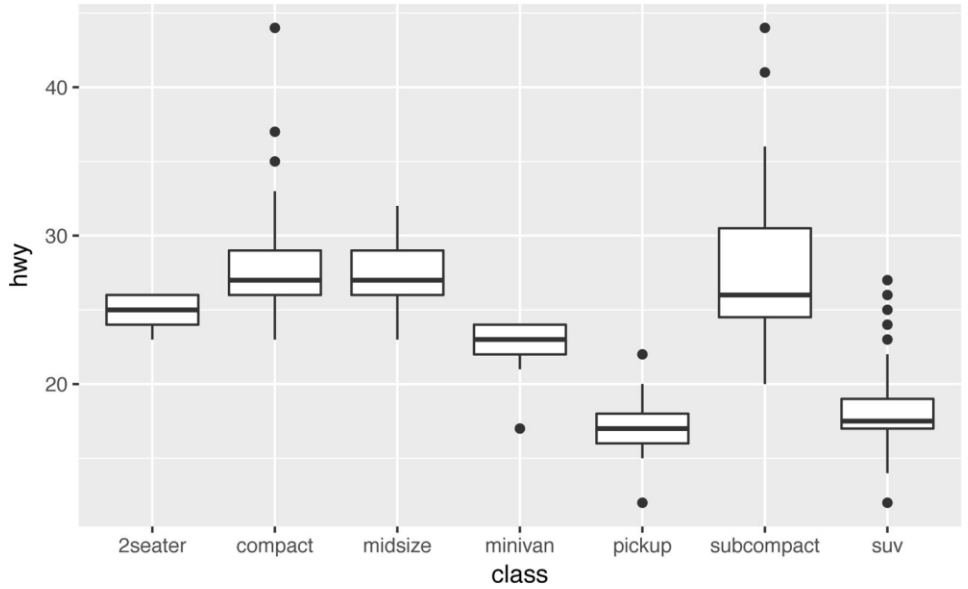
#cut es una variable categórica ordinal (ordenable)

```
ggplot(data = diamonds, mapping = aes(x = cut, y = price)) +
    geom_boxplot()
```



#aquí class es una variable categórica nominal (no ordenable)

```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
    geom_boxplot()
```

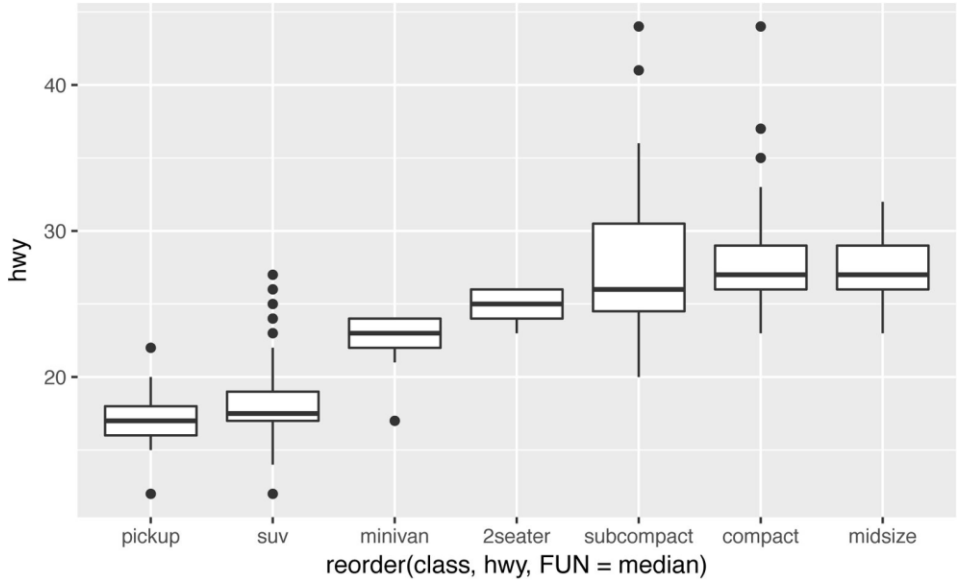


Covariación: categórica con continua

Otra opción para observar la distribución es utilizar **geom_boxplot()**

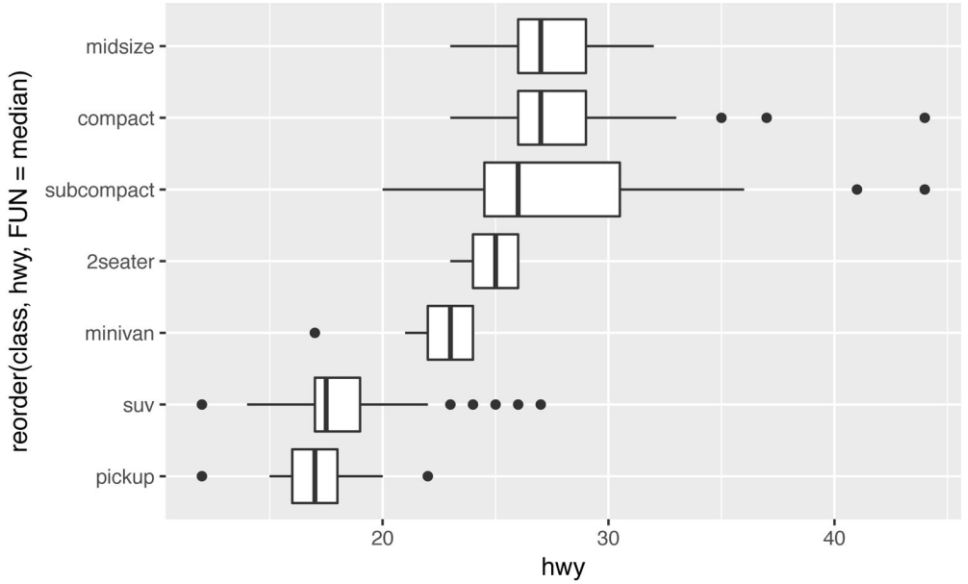
#Aquí se reordena class en función de la mediana de las hwy por cada class

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = reorder(class, hwy,
  FUN = median),y = hwy))
```



#Además, se intercambian los ejes x e y, en este caso para que los nombres largos queden mejor

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = reorder(class, hwy,
  FUN = median),y = hwy)) + coord_flip()
```

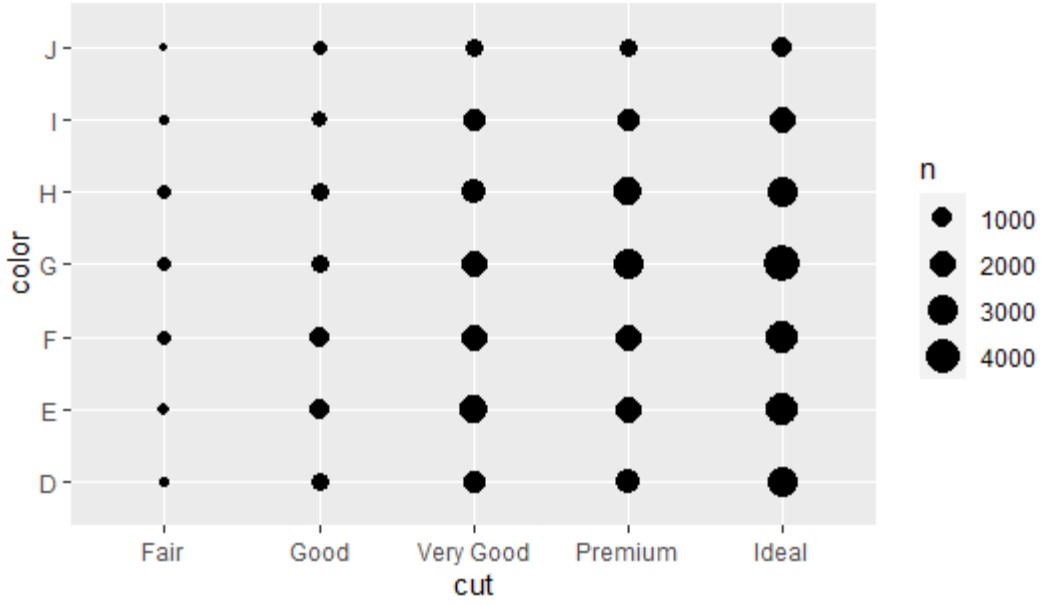


Covariación: categórica con categórica

Se puede utilizar **geom_count()** o directamente **count()**

#Se puede utilizar geom_count()

```
ggplot(data = diamonds) +
    geom_count(mapping = aes(x = cut, y = color))
```



#También se puede utilizar count no "gráfico".

```
diamonds %>% count(color, cut)
```

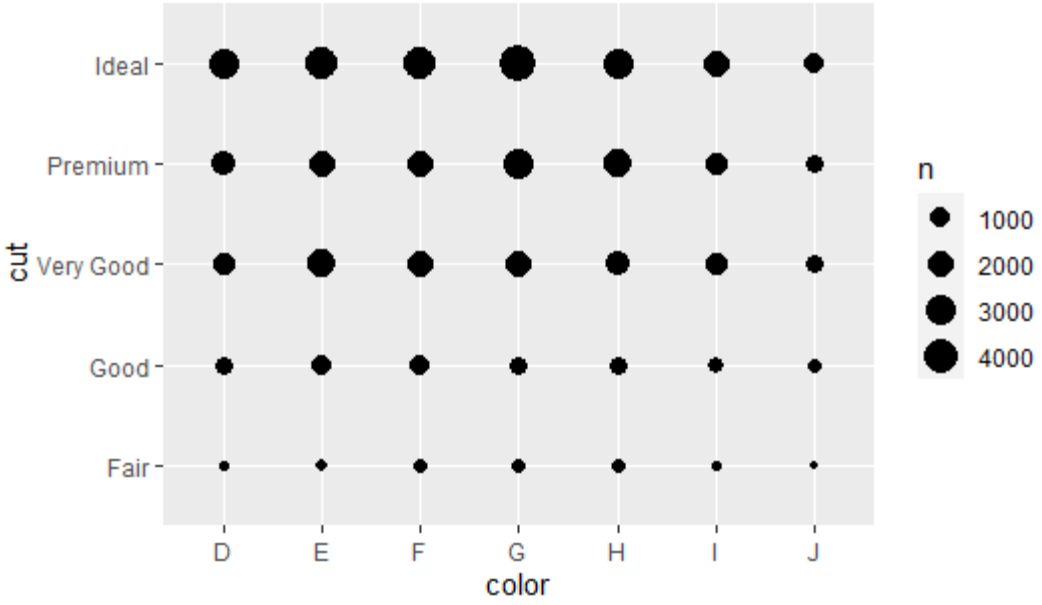
```
A tibble: 35 x 3
   color cut         n
   <ord> <ord>     <int>
 1 D     Fair        163
 2 D     Good        662
 3 D     Very Good  1513
 4 D     Premium    1603
 5 D     Ideal      2834
 6 E     Fair        224
 7 E     Good        933
 8 E     Very Good  2400
 9 E     Premium    2337
10 E     Ideal      3903

# ... with 25 more rows
```

Covariación: categórica con categórica

Se puede utilizar **geom_count()** o **count() + geom_tile()**

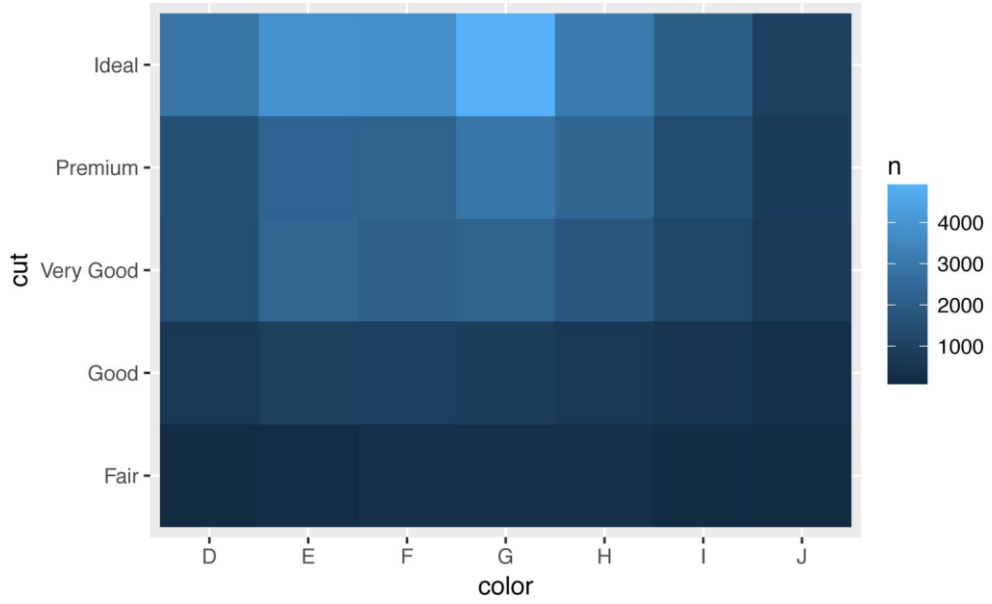
#Se puede utilizar geom_count()

```
ggplot(data = diamonds) +
    geom_count(mapping = aes(x = color, y = cut))
```



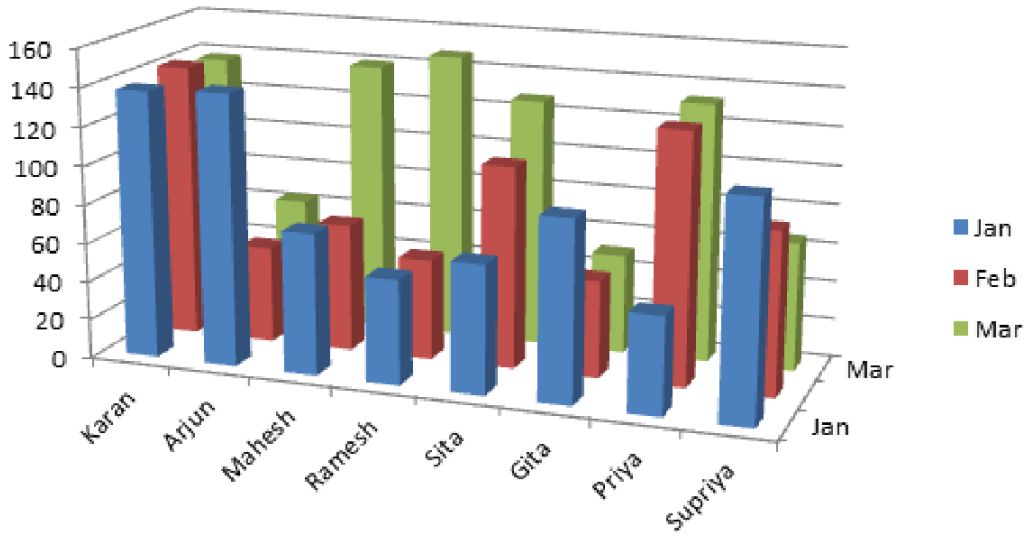
#Otra posibilidad es count() + geom_tile()

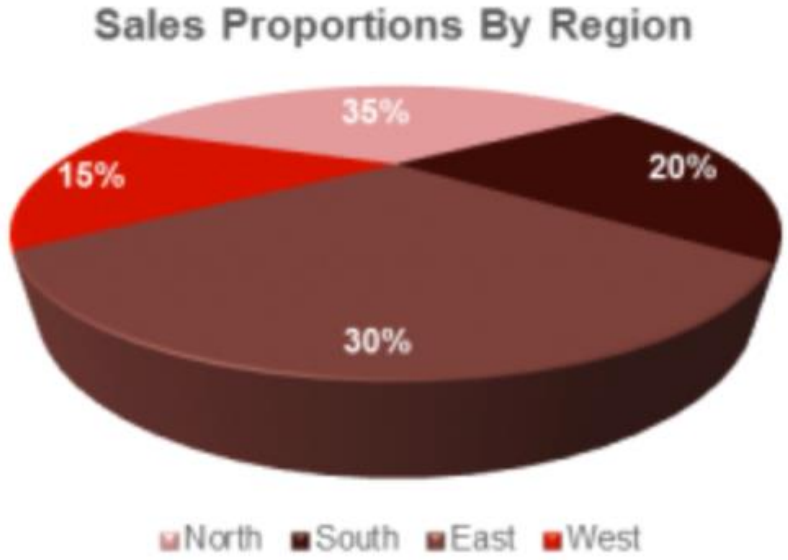
```
diamonds %>% count(color, cut) %>%
    ggplot(mapping = aes(x = color, y = cut)) +
     geom_tile(mapping = aes(fill = n))
```



Covariación: categórica con categórica

¿Por qué no usar un geom_bar3d o algo similar?





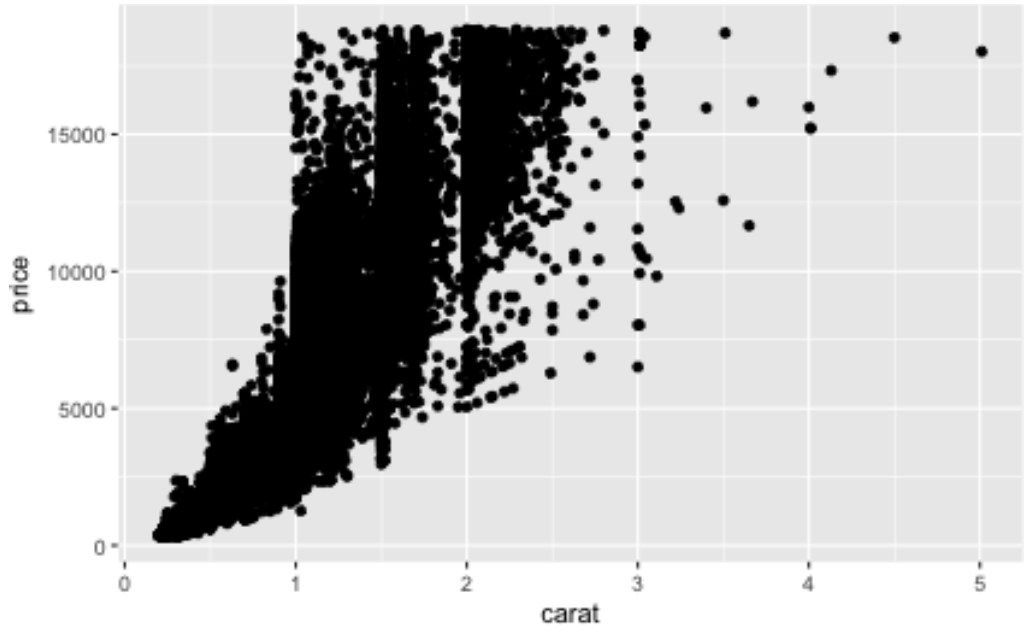
En general son considerados malas prácticas. **Nunca hacer barras ni pie charts en 3D:** https://www.data-to-viz.com/caveat/3d.html. Pero igual hay entusiastas para casos interactivos o animaciones: https://www.tylermw.com/3d-ggplots-with-rayshader/

Covariación: continua con continua

Se puede utilizar **alpha** (transparencia)

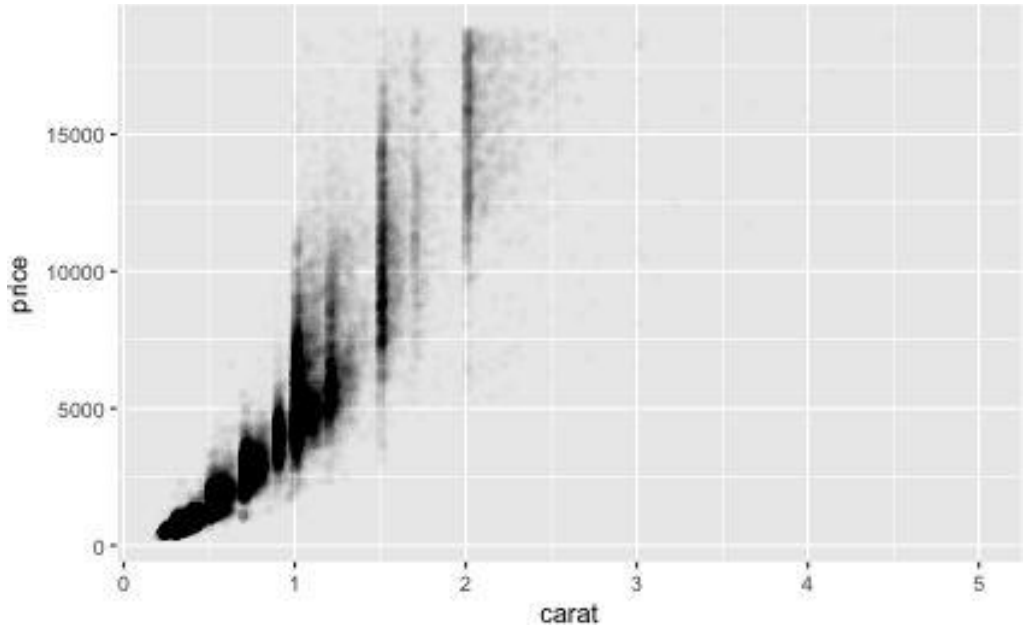
#Si se utiliza geom_point y hay muchos puntos, se pierde noción de la densidad

```
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price))
```



#Se puede utilizar en transparencia, pero no hay una escala al costado, asociada a las tonalidades

```
ggplot(data = diamonds) +
geom_point(mapping = aes(x = carat, y = price), alpha = 1/100)
```

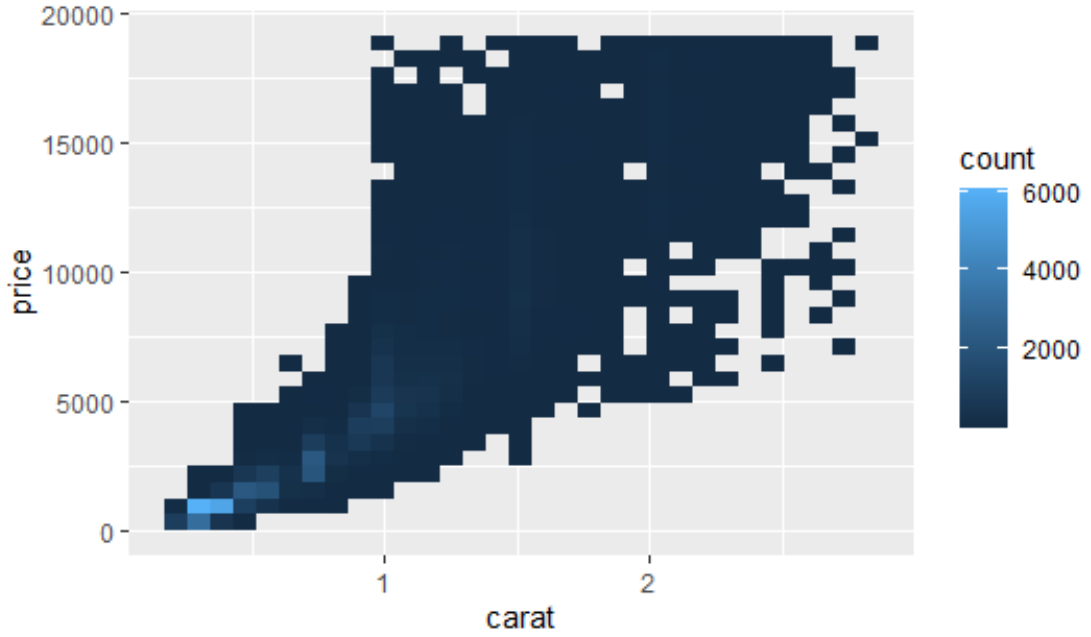


Covariación: continua con continua

Se puede utilizar geom_bin2d(), geom_hex()

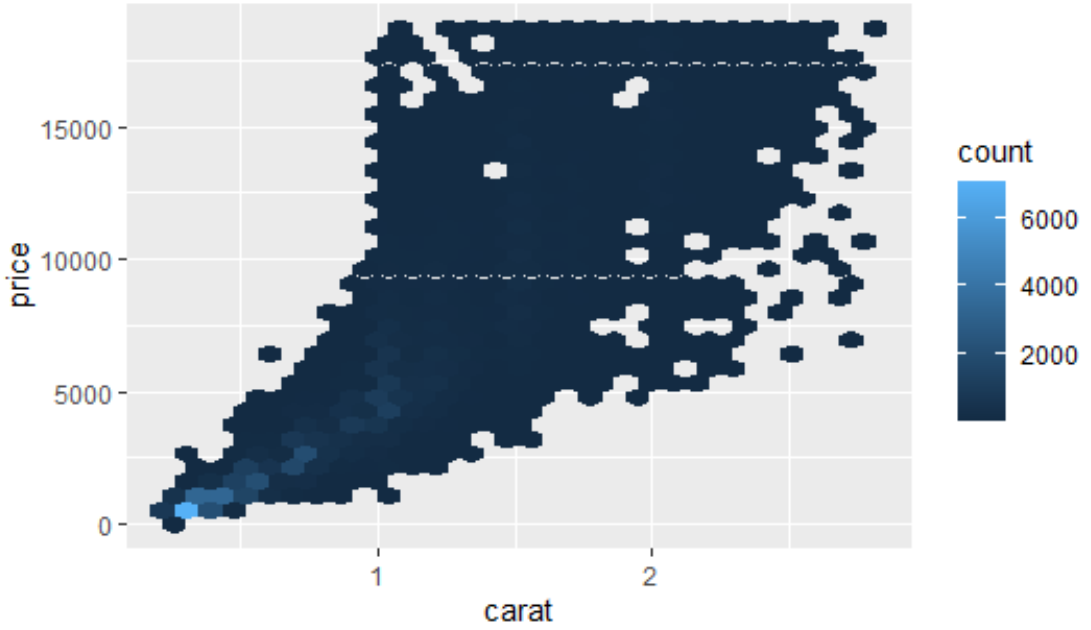
 # geom_bin2d y geom_hex permite manejar mejor la densidad, y ponerle una escala de colores

```
ggplot(data = smaller) +
  geom_bin2d(mapping = aes(x = carat, y = price))
```



#geom_hex es similar a geom_density2d, pero con embaldosado hexagonal.

```
#install.packages("hexbin")
 library(hexbin)
ggplot(data = smaller) +
  geom_hex(mapping = aes(x = carat, y = price))
```

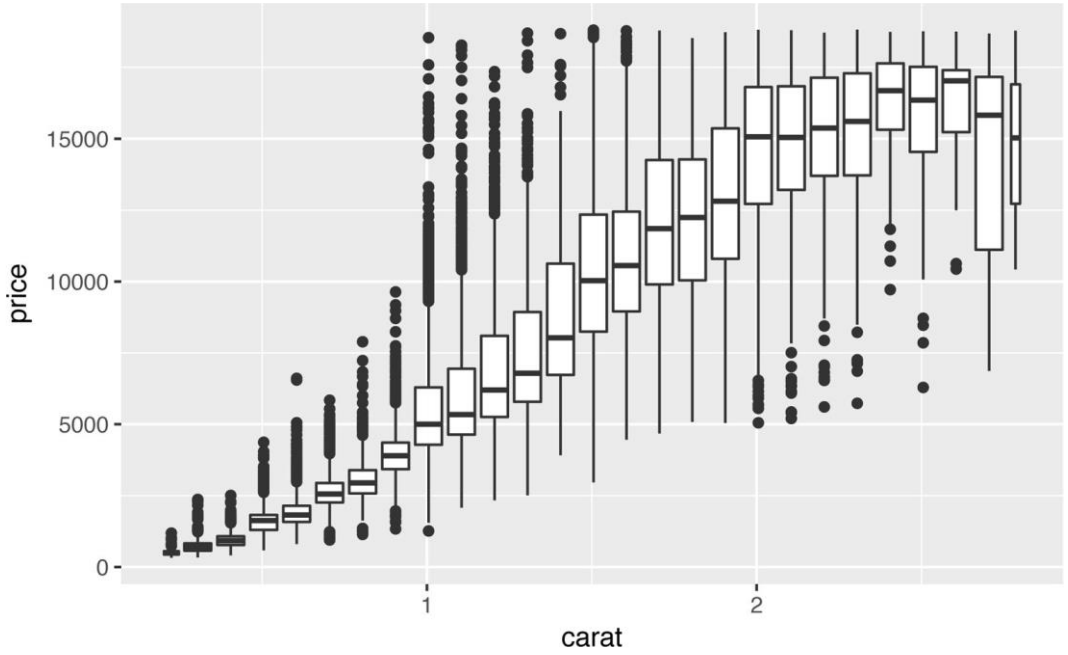


Covariación: continua & continua

Se puede utilizar geom_boxplot() cortando por secciones o por igual cantidad de obs.

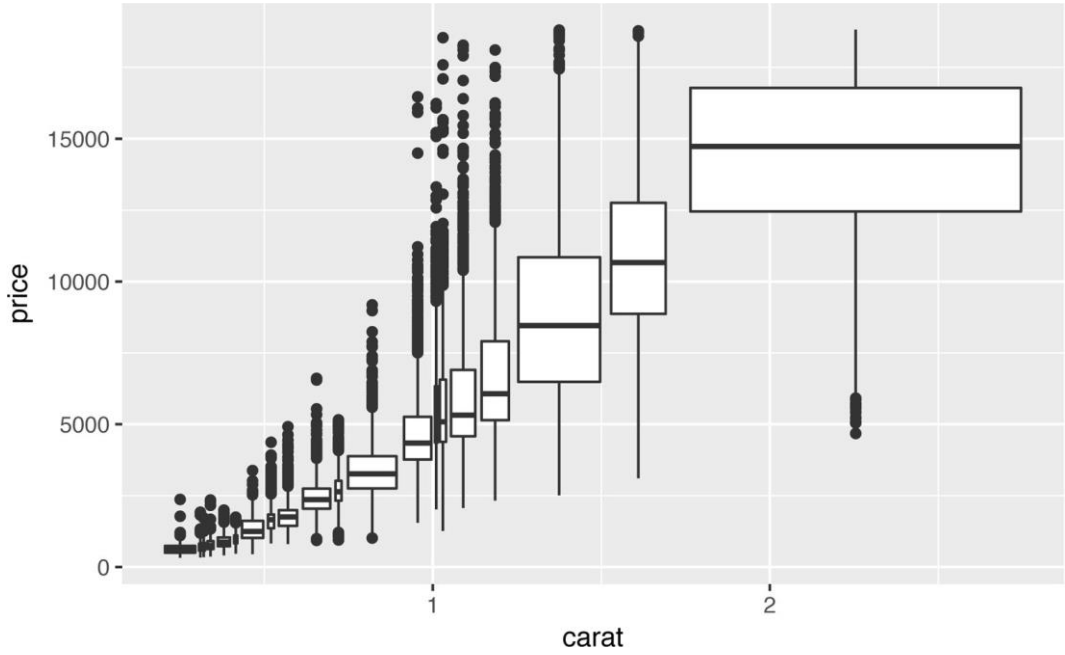
#geom_boxplot también puede utilizarse, dividiendo una de ellas en secciones o grupos

```
ggplot(data = smaller, mapping = aes(x = carat, y = price)) +
geom_boxplot(mapping = aes(group = cut_width(carat,0.1)))
```



#geom_boxplot pero manteniendo la misma cantidad de observaciones por boxplot()

```
ggplot(data = smaller, mapping = aes(x = carat, y = price)) +
geom_boxplot(mapping = aes(group = cut_number(carat, 20)))
```



Patrones y Modelos (una introducción)

Si en una relación entre dos variables hay un patrón visual en los datos, ¿qué debemos preguntarnos?

Preguntas posibles:

- ¿Podría deberse a una coincidencia?
- ¿Cómo podría describirse esa relación implicada en el patrón?
- ¿Cuán fuerte es esa relación visualizada en el patrón?
- ¿Qué otras variables pueden afectar esa relación?
- ¿Puede esa relación cambiar si se observan individualmente a subgrupos de los datos?

Patrones y Modelos (una introducción)

Duración de 272 erupciones y el tiempo entre ellas del Geiser Old Faithful en Yellowstone National Park. Se visualiza un patrón entre eruption y waiting.

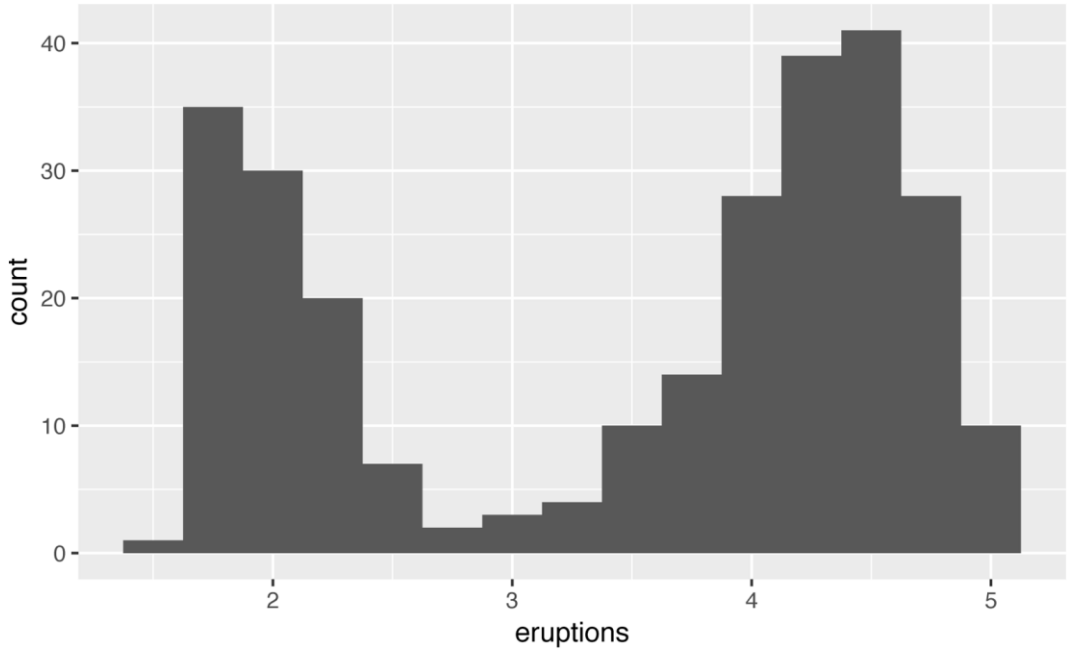
```
> faithful
    eruptions waiting
1       3.600      79
2       1.800      54
3       3.333      74
4       2.283      62
5       4.533      85
6       2.883      55
7       4.700      88
8       3.600      85
9       1.950      51
10      4.350      85
:
269     2.150      46
270     4.417      90
271     1.817      46
272     4.467      74
```

Patrones y Modelos (una introducción)

Duración de 272 erupciones y el tiempo entre ellas del Geiser Old Faithful en Yellowstone National Park. Se visualiza un patrón entre eruption y waiting.

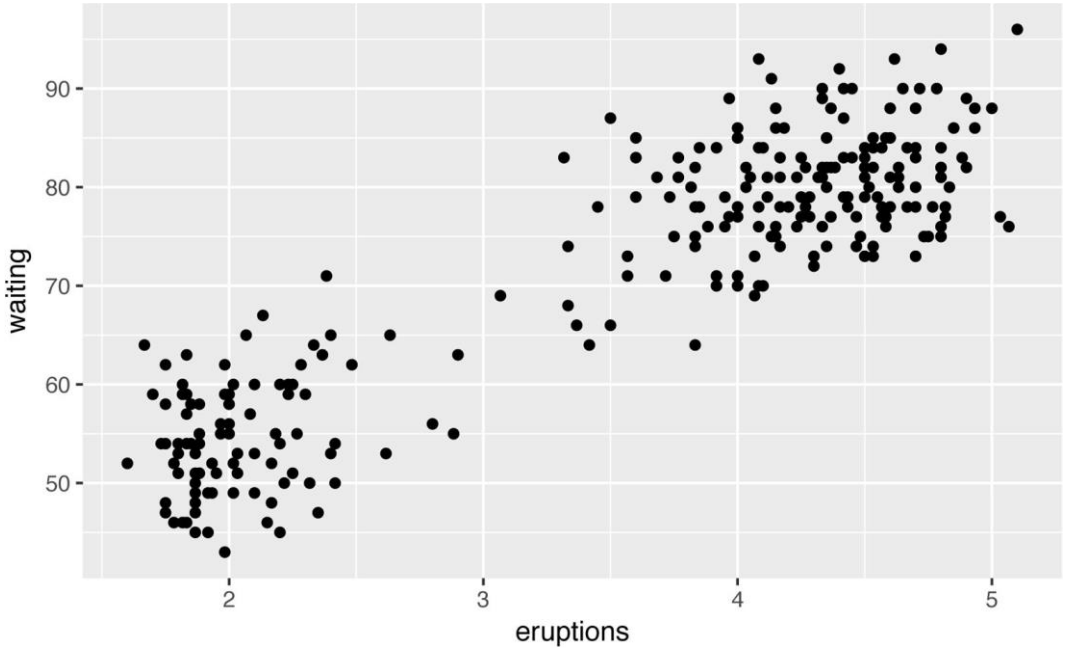
#Se perciben 2 cluster de datos.

```
ggplot(data = faithful, mapping = aes(x = eruptions)) +
  geom_histogram(binwidth = 0.25)
```



 # Se perciben 2 clusters en la correlación entre 2 variables

```
ggplot(data = faithful) +
  geom_point(mapping = aes(x = eruptions, y = waiting))
```

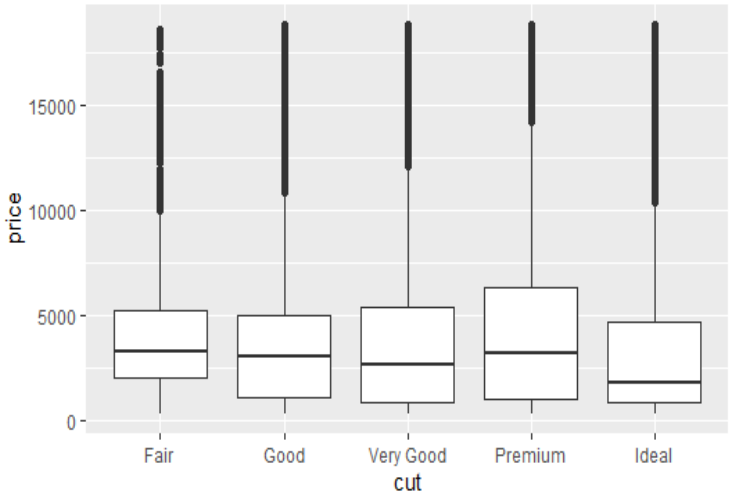


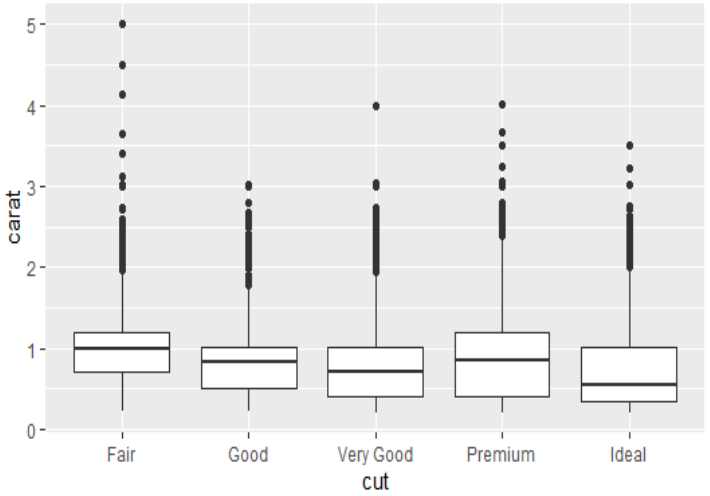
Patrones y Modelos (una introducción)

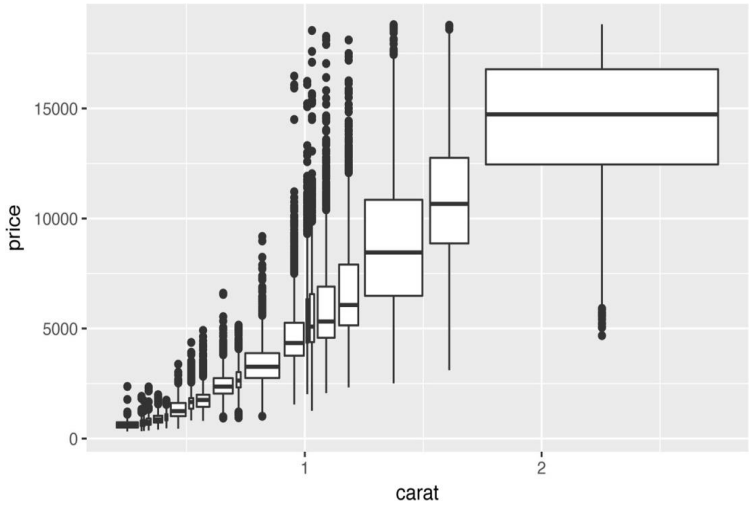
- Si la variación de un fenómeno produce incertidumbre, la covariación la reduce.
- El valor de una variable puede producir una predicción más ajustada de otra variable.
- Si hay una relación causal entre las variables, se puede utilizar el valor de una variable para controlar el valor de la otra variable.

Patrones y Modelos (una introducción)

- Los modelos son herramientas para extraer patrones en los datos.
- Ejemplo: En diamonds, **es difícil extraer una "explicación" a la relación entre *cut* y *price***, porque influyen otras relaciones, sobre todo la relación entre *carat* y *price*.







Patrones y Modelos (una introducción)

Se analiza la relación entre *cut* y *price*, pero minimizando la relación entre cut y carat:

- Primero se calcula la relación lineal entre **log(*price*)** y **log(*carat*)**.
- **log(*price*)** = A + B***log(*carat*)** + **resid**
- ***price*** = exp(A + B***log(*carat*)**) * **exp(resid)**
- Luego dibujamos la curva y la comparamos con los datos (para comprobar).
- Luego se compara el exponente del residuo -**exp(resid)**- con respecto al cut.

#Primero se calcula el modelo lineal entre log(price) y log(carat)

```
library(modelr)
mod <- lm(log(price) ~ log(carat), data = smaller)        #?lm para entender un poco más
```

#hacer > mod para ver la información contenida #Luego se agrega el residuo a smaller (diamantes con carat <3)

```
smaller <- smaller %>%  add_residuals(mod) %>%                    #?add_residuals
        mutate(resid = exp(resid));                               #mutate
mod$coefficients[1]                    #coeficiente A    [intercept]
mod$coefficients[2]                    #coeficiente B    [log(carat)]
```

#Dibujamos la curva que aproxima a la distribución de carat y precio.

```
xx <- seq(0.1,2.5,0.1)
yy <- exp(mod$coefficients[1] + mod$coefficients[2]*log(xx))   # xx e yy para dibujar la reg. lineal
ggplot() +
  geom_point(data=smaller,mapping = aes(y=price,x=carat), alpha=0.01) +
  geom_line(mapping = aes(y=yy,x=xx), color= "red", size=1.2) +          #curva de regresión lineal
  geom_smooth(data=smaller , mapping = aes(y=price,x=carat));           #curva de geom_smooth()
```

Patrones y Modelos (una introducción)

Luego de “anular” la relación entre carat y precio, queda más claro que hay una relación creciente entre cut y precio.