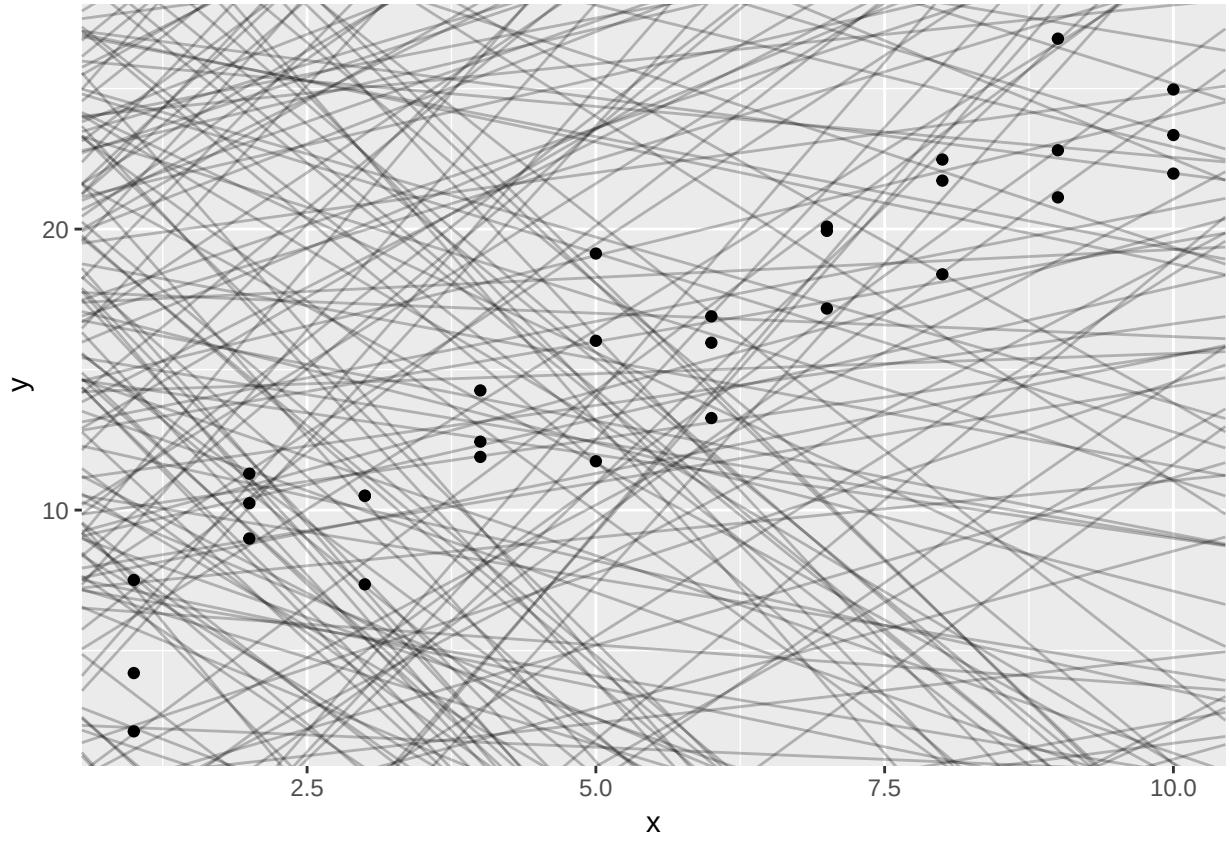
Parte IV: Modelado

ronny hdez-mora

5/26/2019

```
models <- tibble(
  a1 = runif(250, -20, 40),
  a2 = runif(250, -5, 5)
)

ggplot(sim1, aes(x, y)) +
  geom_abline(
    aes(intercept = a1, slope = a2),
    data = models, alpha = 1/4
  ) +
  geom_point()
```



Los 250 modelos anteriores hay unos que son muy malos. Necesitamos uno que esté más cerca de los datos. Podemos ajustar uno encontrando los valores de `a_0` y `a_1` que genere el modelo con las distancias mínimas a estos datos.

Esto se puede hacer con la distancia vertical entre cada punto y el modelo. Esta distancia es la diferencia entre el valor de y dado por el modelo (predicción) y el valor de y real en los datos.

```
model1 <- function(a, data) {
  a[1] + data$x * a[2]
}

model1(c(7, 1.5), sim1)
```

```
##  [1]  8.5  8.5  8.5 10.0 10.0 10.0 11.5 11.5 11.5 13.0 13.0 13.0 14.5 14.5
## [15] 14.5 16.0 16.0 16.0 17.5 17.5 17.5 19.0 19.0 19.0 20.5 20.5 20.5 22.0
## [29] 22.0 22.0
```

... Revisar este capítulo después

Capítulo 19

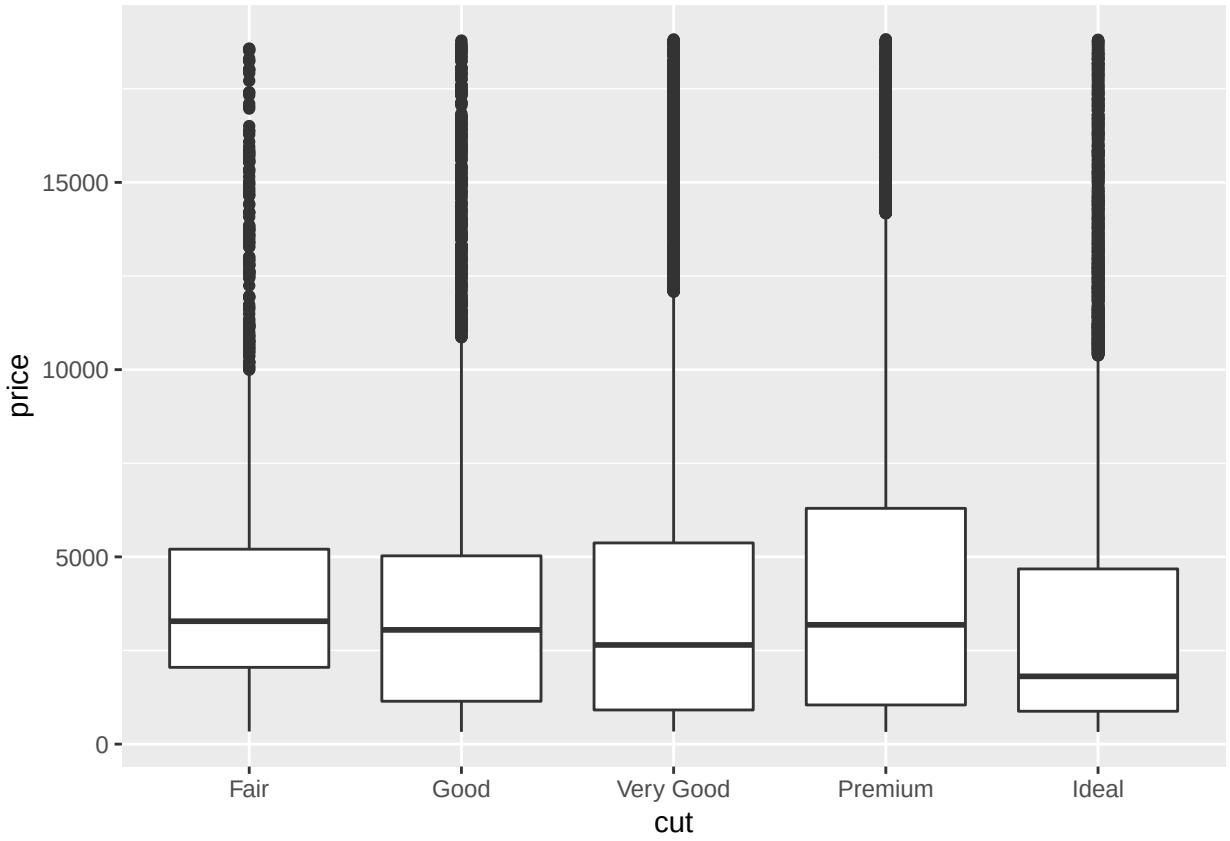
```
library(nycflights13)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

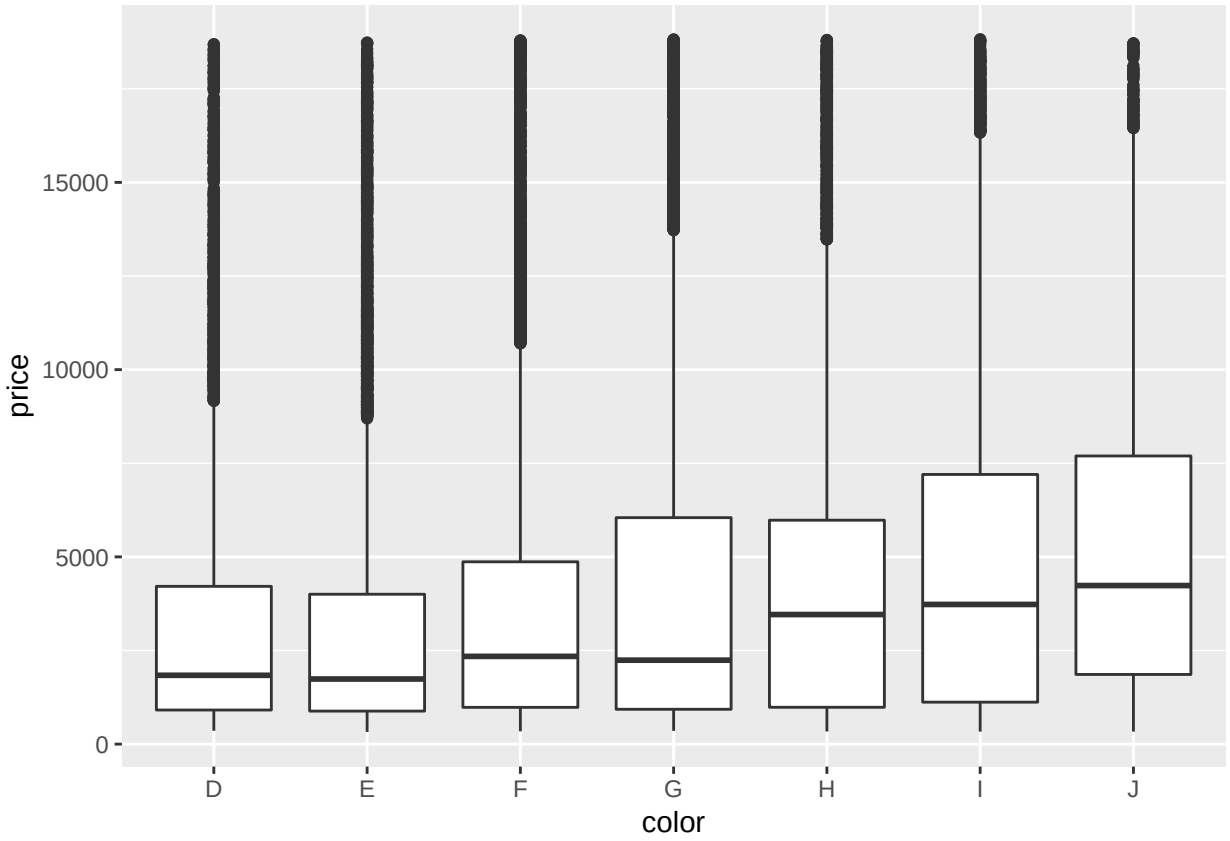
```
## The following object is masked from 'package:base':
##
##     date
```

¿Porqué son los diamantes de baja calidad caros?

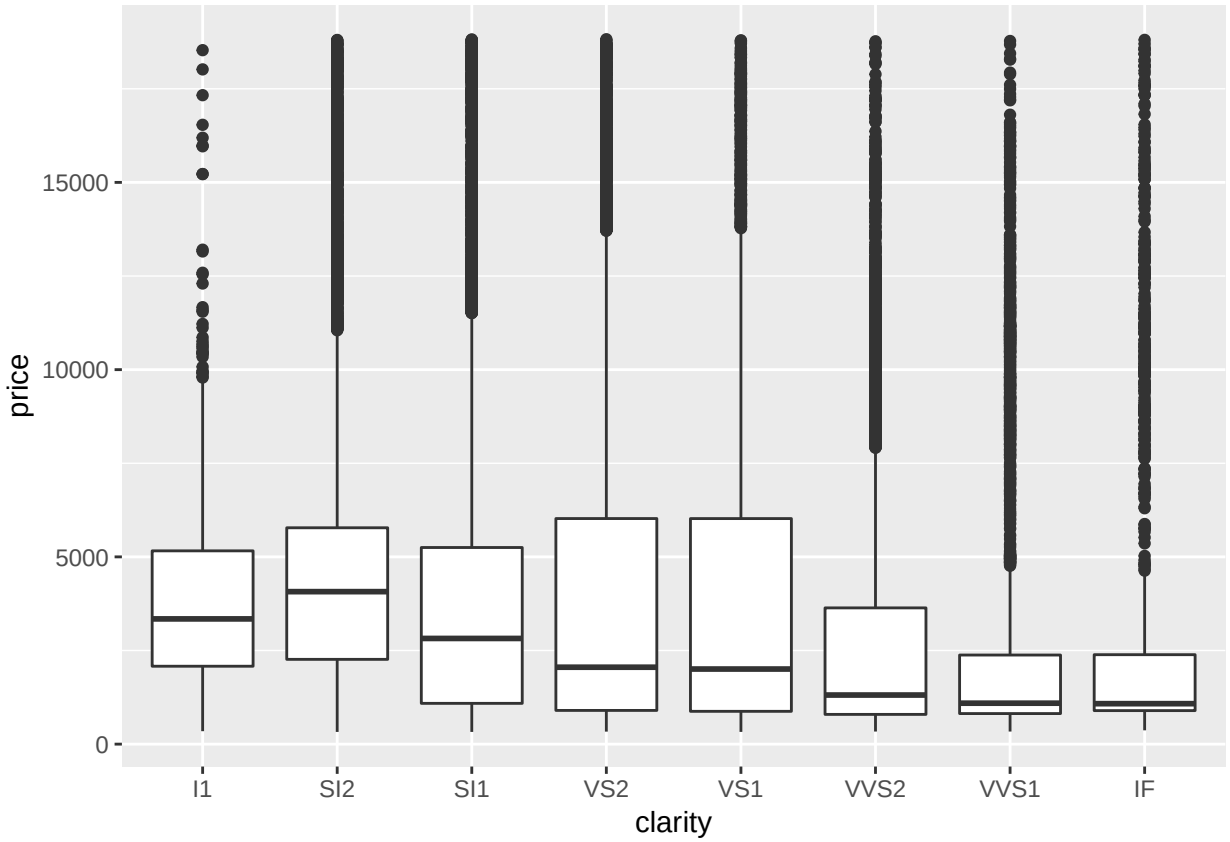
```
ggplot(diamonds, aes(cut, price)) +
  geom_boxplot()
```




```
ggplot(diamonds, aes(color, price)) +
  geom_boxplot()
```

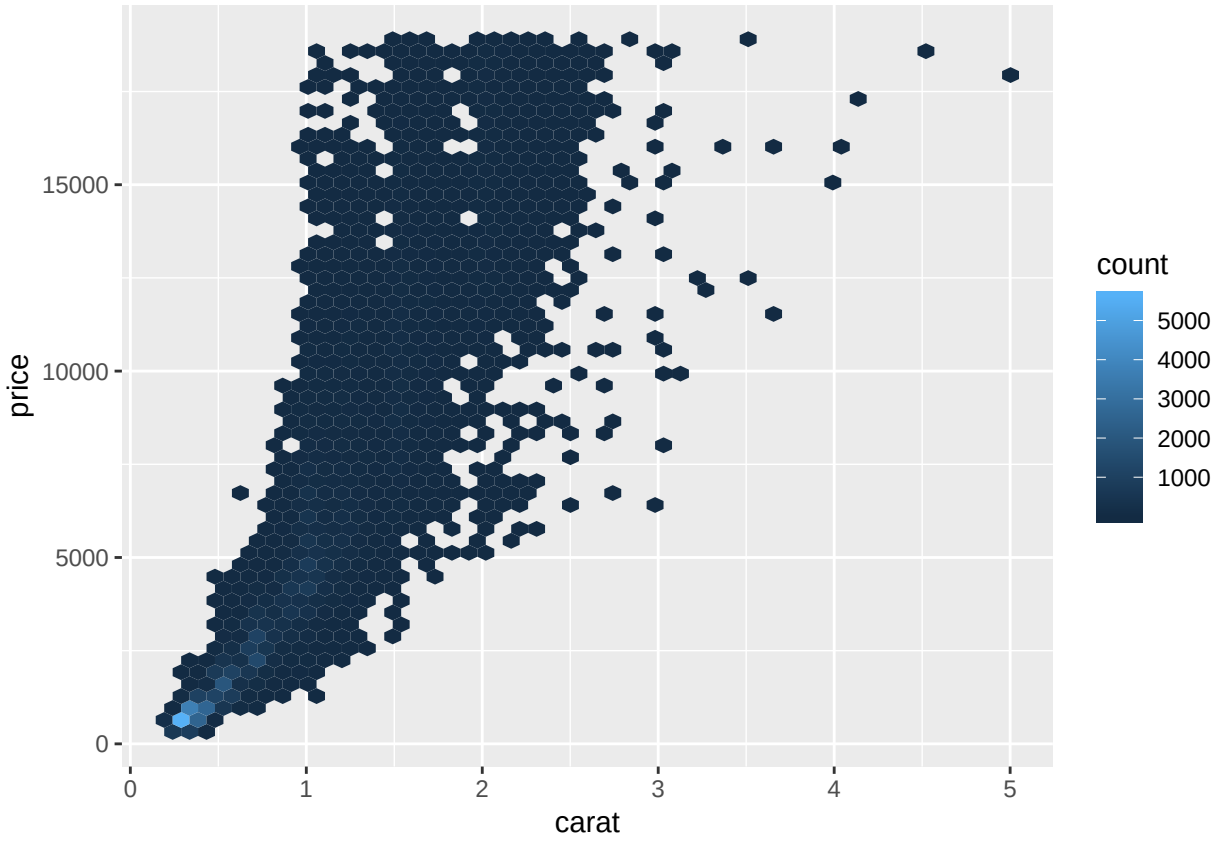



```
ggplot(diamonds, aes(clarity, price)) +
  geom_boxplot()
```



Lo anterior muestra que características malas parecen tener precios más altos. Sin embargo hay uyna característica que está relacionada con el precio y es el peso (carat)

```
ggplot(diamonds, aes(carat, price)) +
  geom_hex(bins = 50)
```



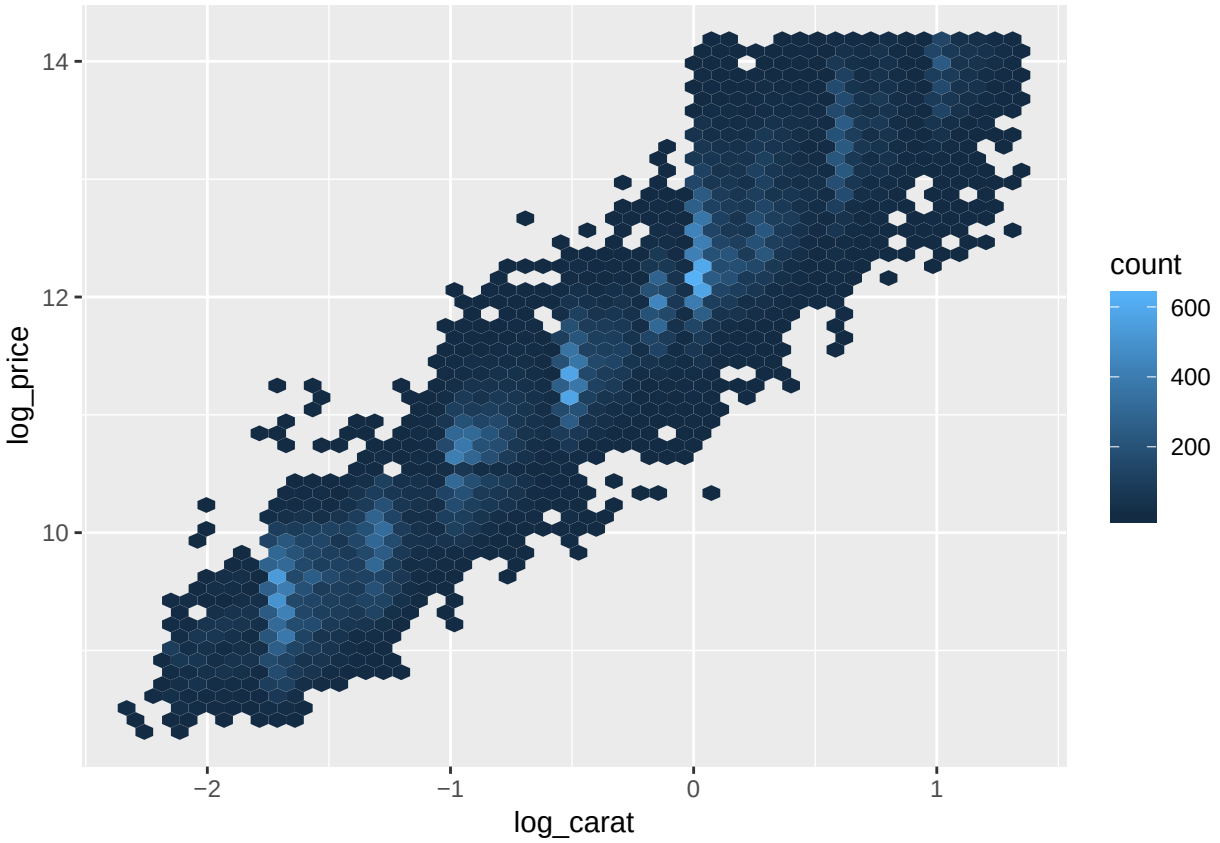
Vamos a limpiar los datos para hacerlos un poco más manipulables:

- Enfocarse en aquellos menores a 2.5 de carat
- Hacer transformación logaritmica de carat y price

```
diamonds2 <- diamonds %>%
  filter(carat <= 2.5) %>%
  mutate(log_price = log2(price),
         log_carat = log2(carat))
```

Con esos cambios la relación será más fácil de verla entre carat y price

```
ggplot(diamonds2, aes(x = log_carat, y = log_price)) +
         geom_hex(bins = 50)
```

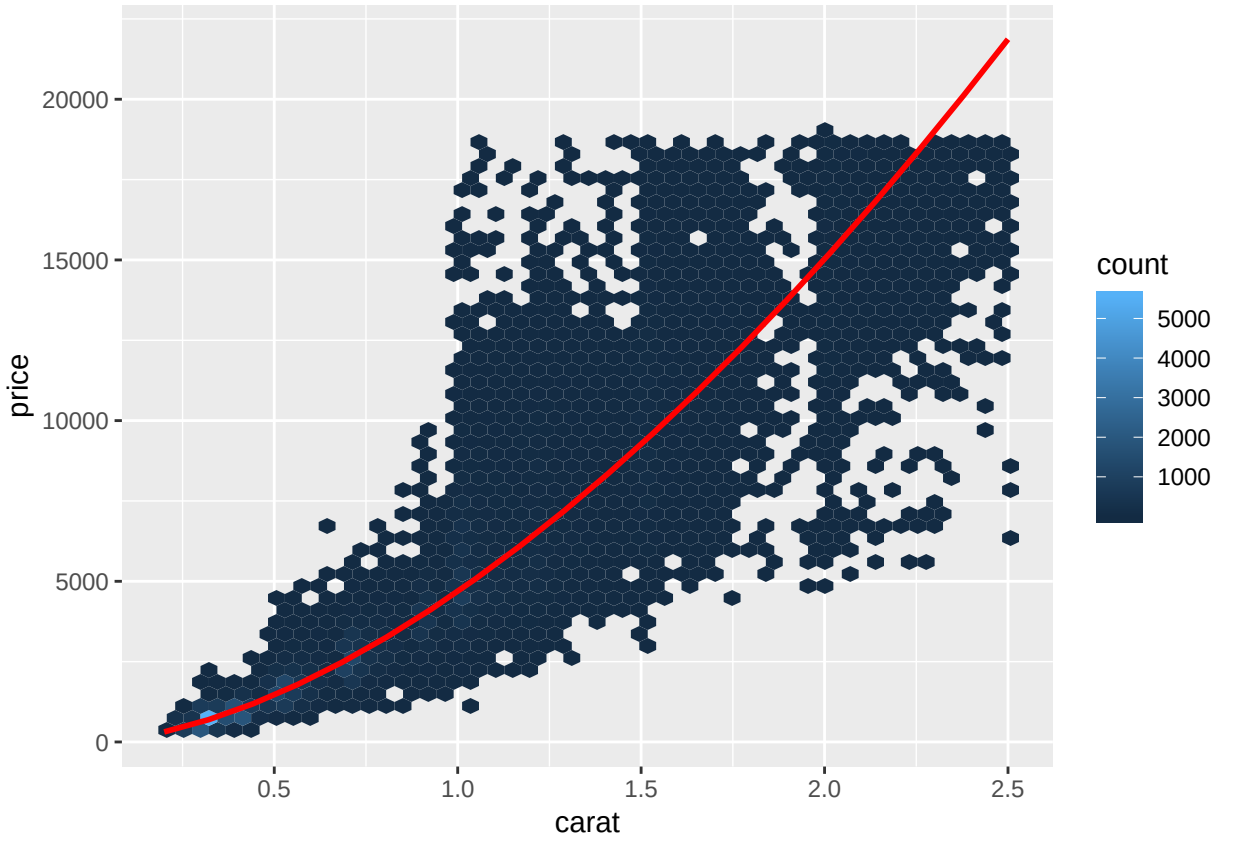



```
mod_diamond <- lm(log_price ~ log_carat, data = diamonds2)
```

Vamos a sobreponer las predicciones del modelo en el gráfico con los valores iniciales

```
grid <- diamonds2 %>%
  data_grid(carat = seq_range(carat, 20)) %>%
  mutate(log_carat = log2(carat)) %>%
  add_predictions(mod_diamond, "log_price") %>%
  mutate(price = 2 ^ log_price)

ggplot(diamonds2, aes(carat, price)) +
  geom_hex(bins = 50) +
  geom_line(data = grid, color = "red", size = 1)
```

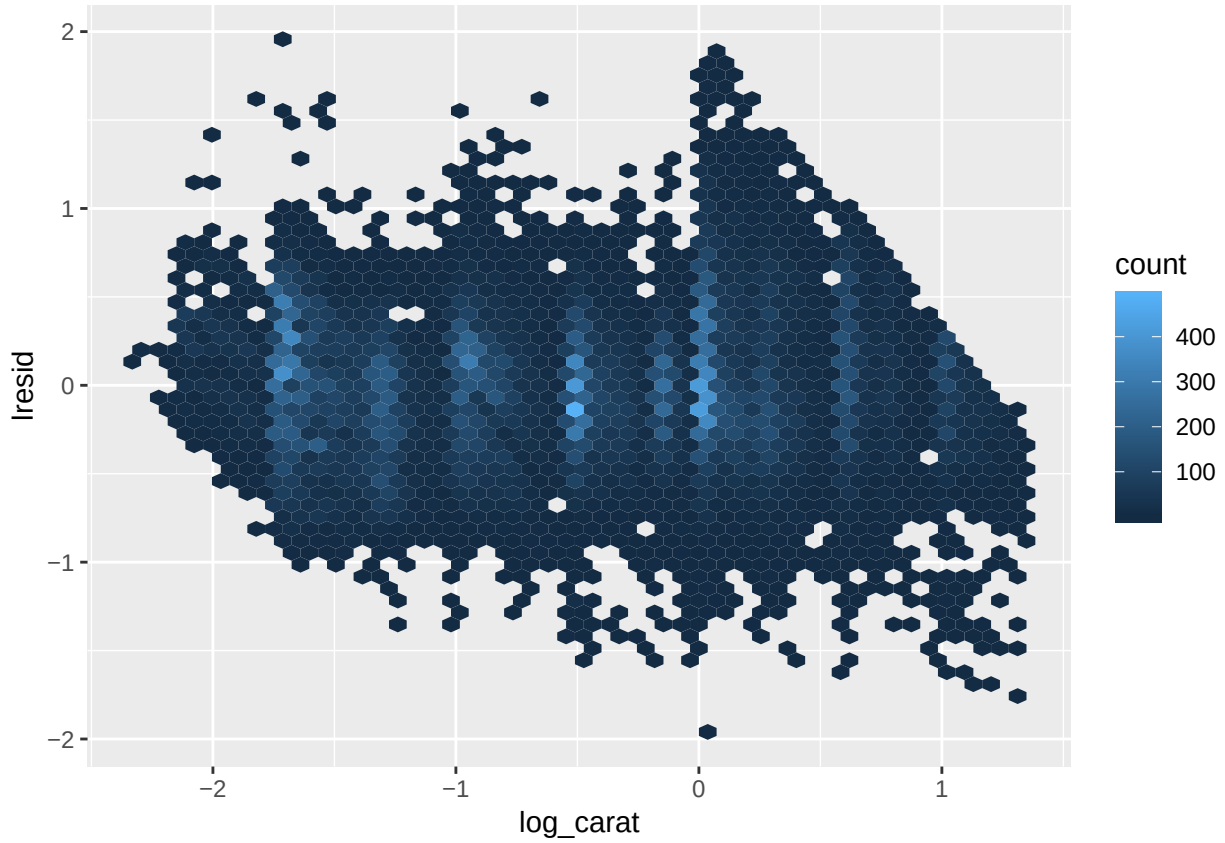


Si creemos en nuestro modelo, lo qu vemos es que los diamantes grandes son mucho más baratos de lo esperado.

Vamos a revisar los residuales para corroborar que se ha elimiado el patrón lineal

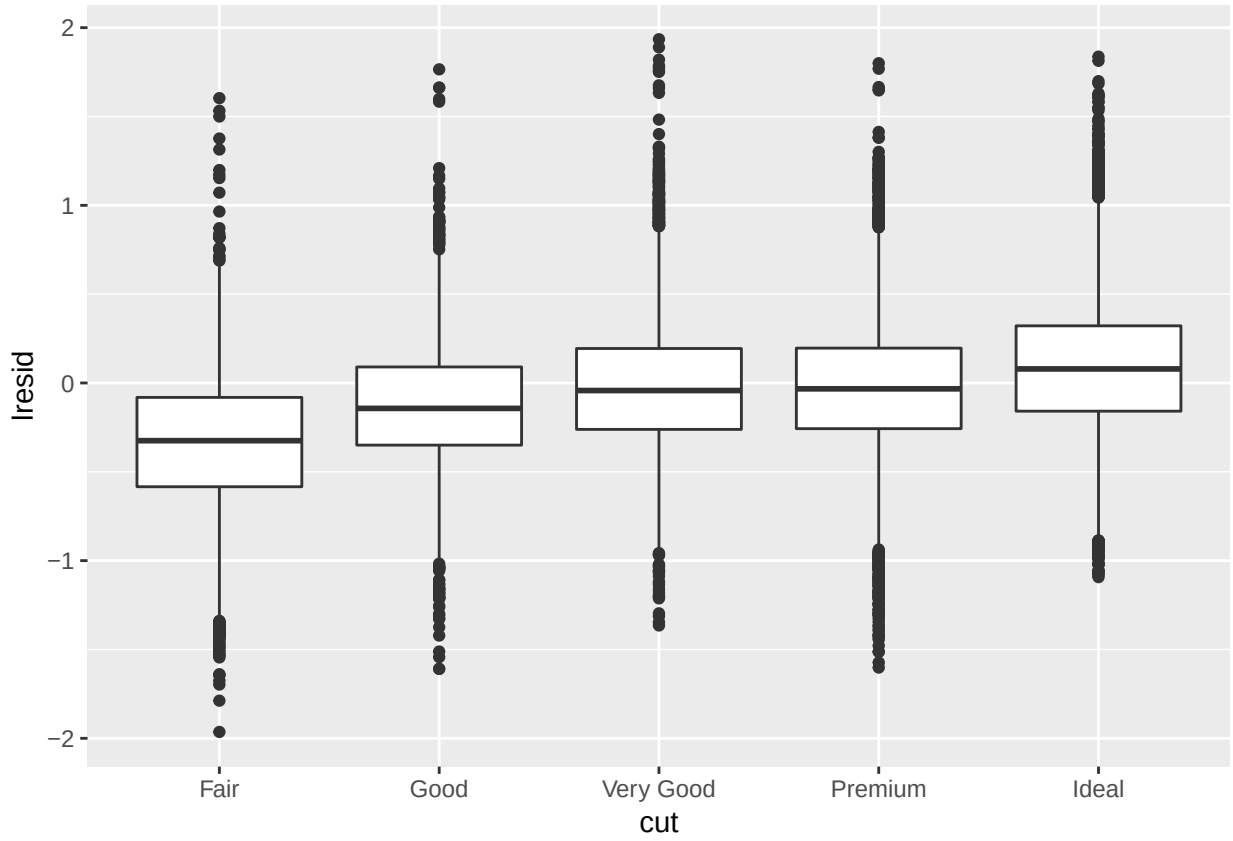
```
diamonds2 <- diamonds2 %>%
  add_residuals(mod_diamond, "lresid")

ggplot(diamonds2, aes(log_carat, lresid)) +
  geom_hex(bins = 50)
```

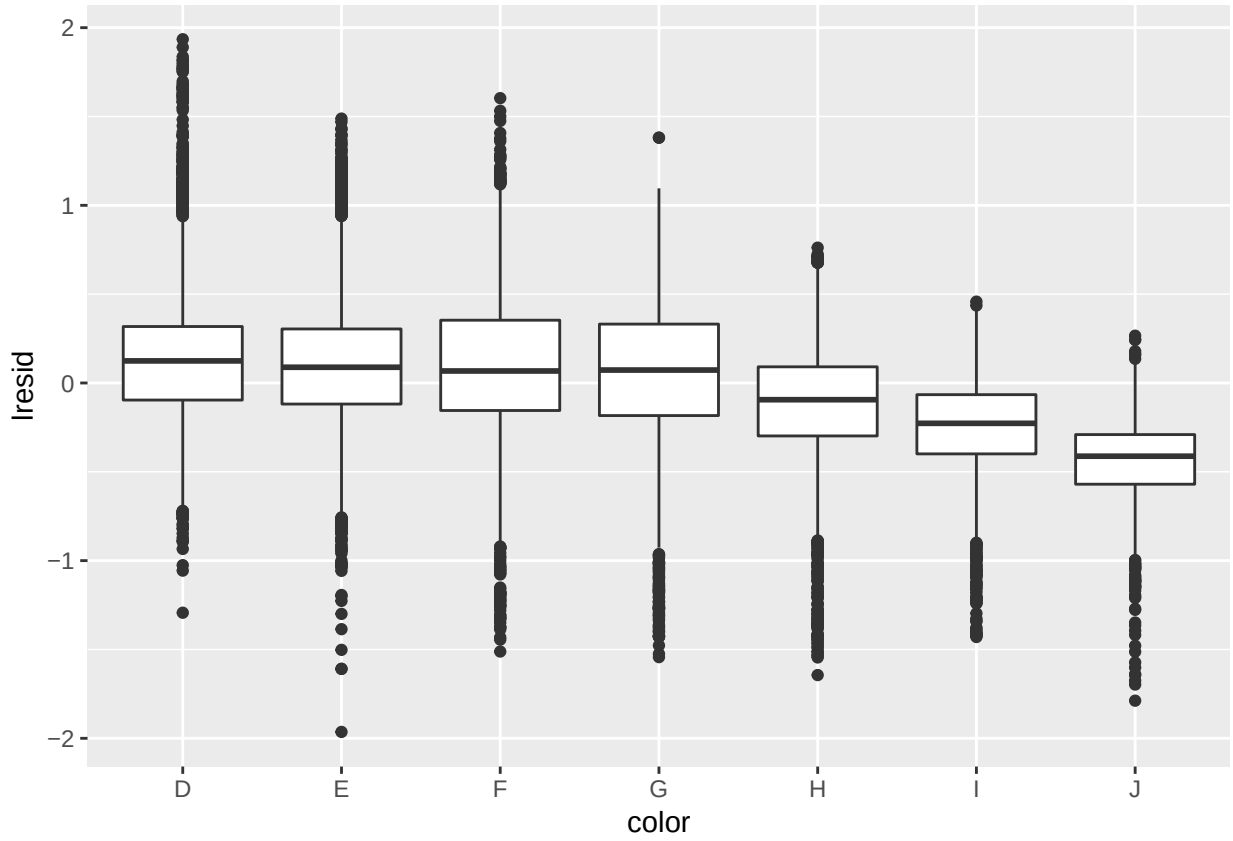


Ahora podemos hacer nuestros gráficos iniciales usando los residuales en lugar del precio

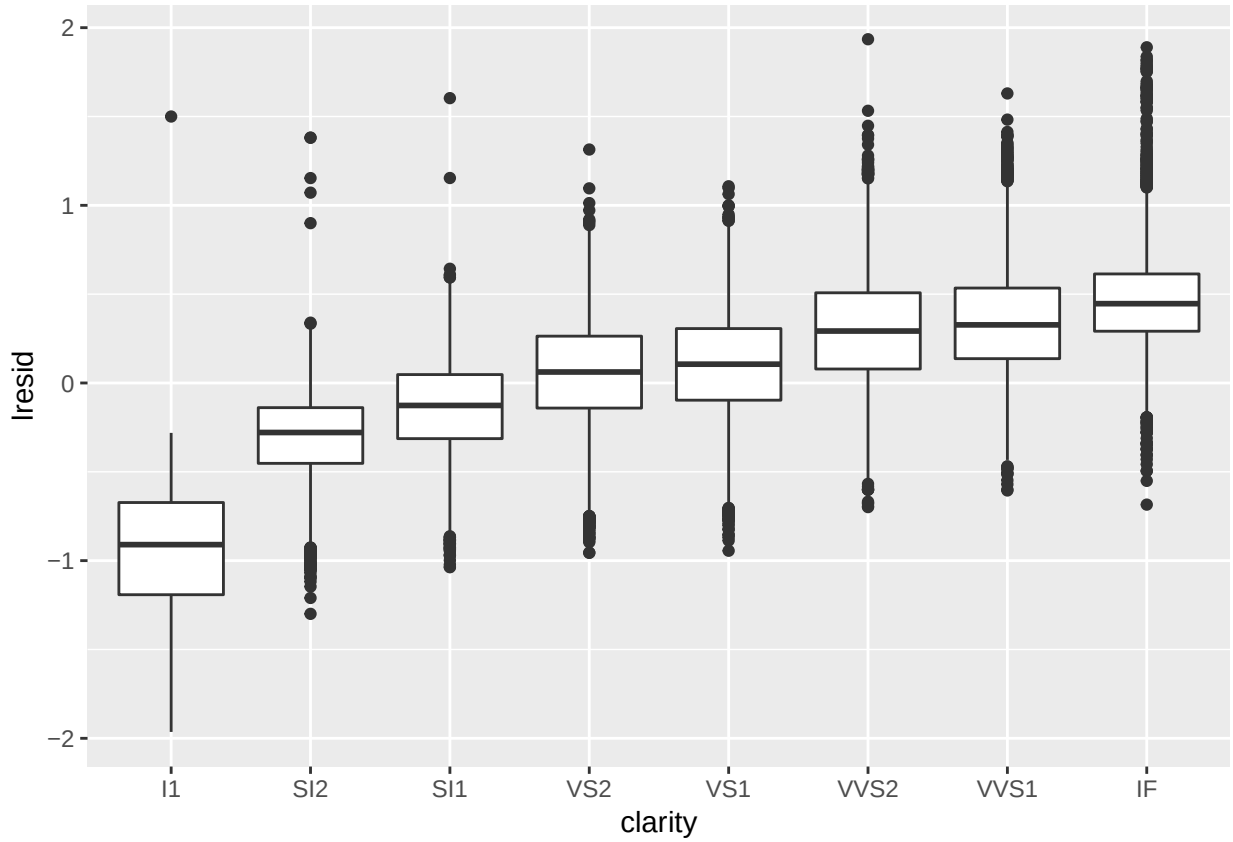
```
ggplot(diamonds2, aes(cut, lresid)) +
  geom_boxplot()
```




```
ggplot(diamonds2, aes(color, lresid)) +
  geom_boxplot()
```




```
ggplot(diamonds2, aes(clarity, lresid)) +
  geom_boxplot()
```



Ahora vemos que a mayor calidad del diamante, mayor el precio.