

Tarea13

Yimmy Eman

2022-07-10

Pregunta 1

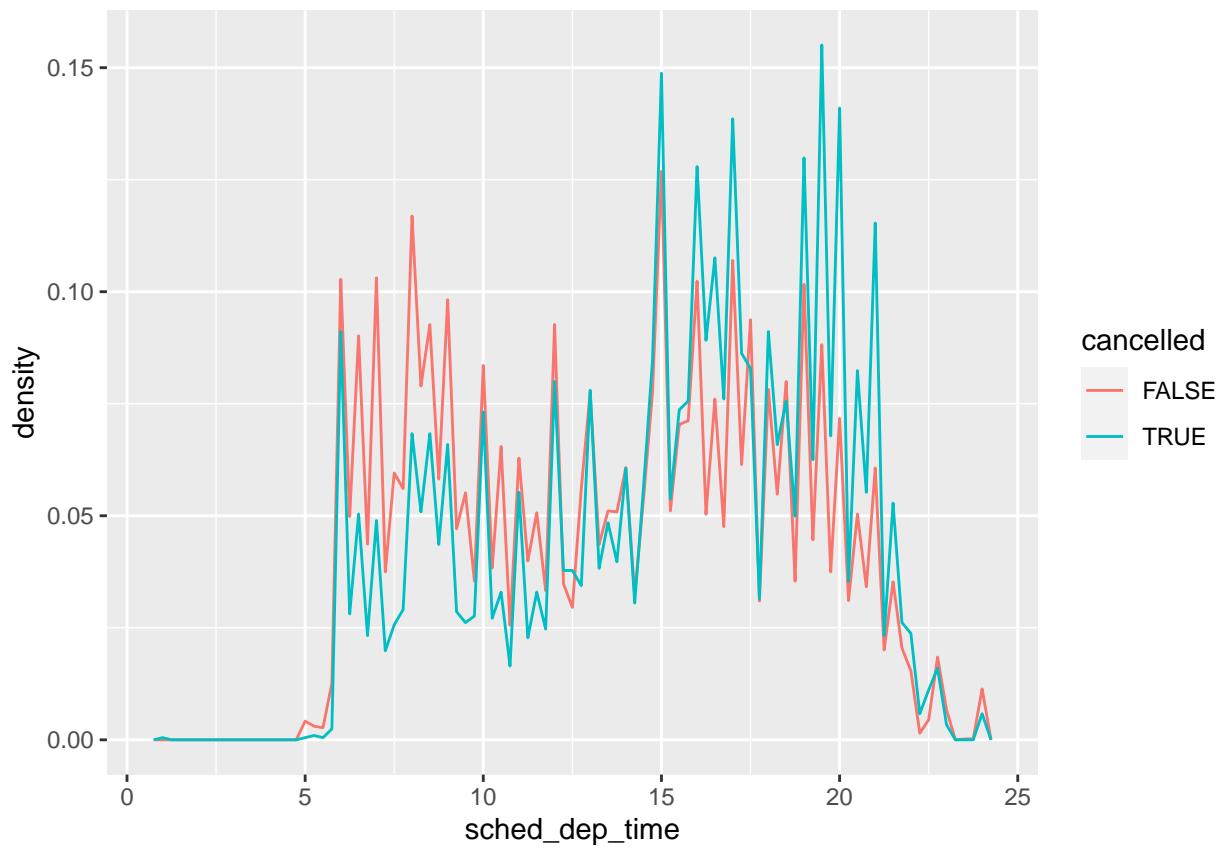
Es hora de aplicar todo lo que hemos aprendido para visualizar mejor los tiempos de salida para vuelos cancelados vs los no cancelados.

Recuerda bien qué tipo de dato tenemos en cada caso.

¿Qué deduces acerca de los retrasos según la hora del día a la que está programada el vuelo de salida?

R: Los vuelos entre las 17:00 y 21:00 aproximadamente tienden a ser cancelados.

```
flights %>%
  mutate(
    cancelled = is.na(dep_time),
    sched_hour = sched_dep_time %% 100, # Devuelve horas
    sched_min = sched_dep_time %% 100, # Devuelve minutos
    sched_dep_time = sched_hour + sched_min / 60 # Todo en formato horas
  ) %>%
  ggplot(aes(sched_dep_time)) +
  geom_freqpoly(aes(y = ..density.., color = cancelled),
                binwidth = 1/4)
```

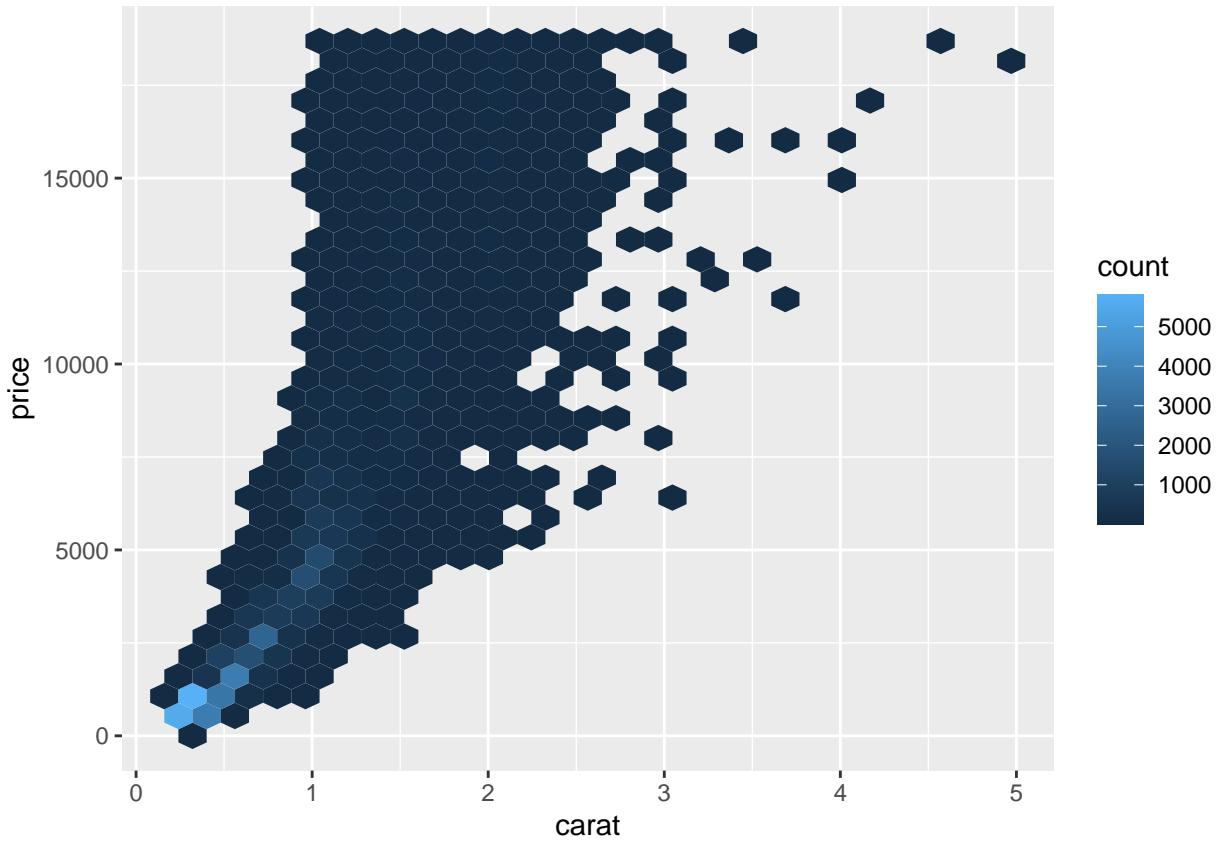


Pregunta 2

- ¿Qué variable del dataset de diamantes crees que es la más importante para poder predecir el precio de un diamante?

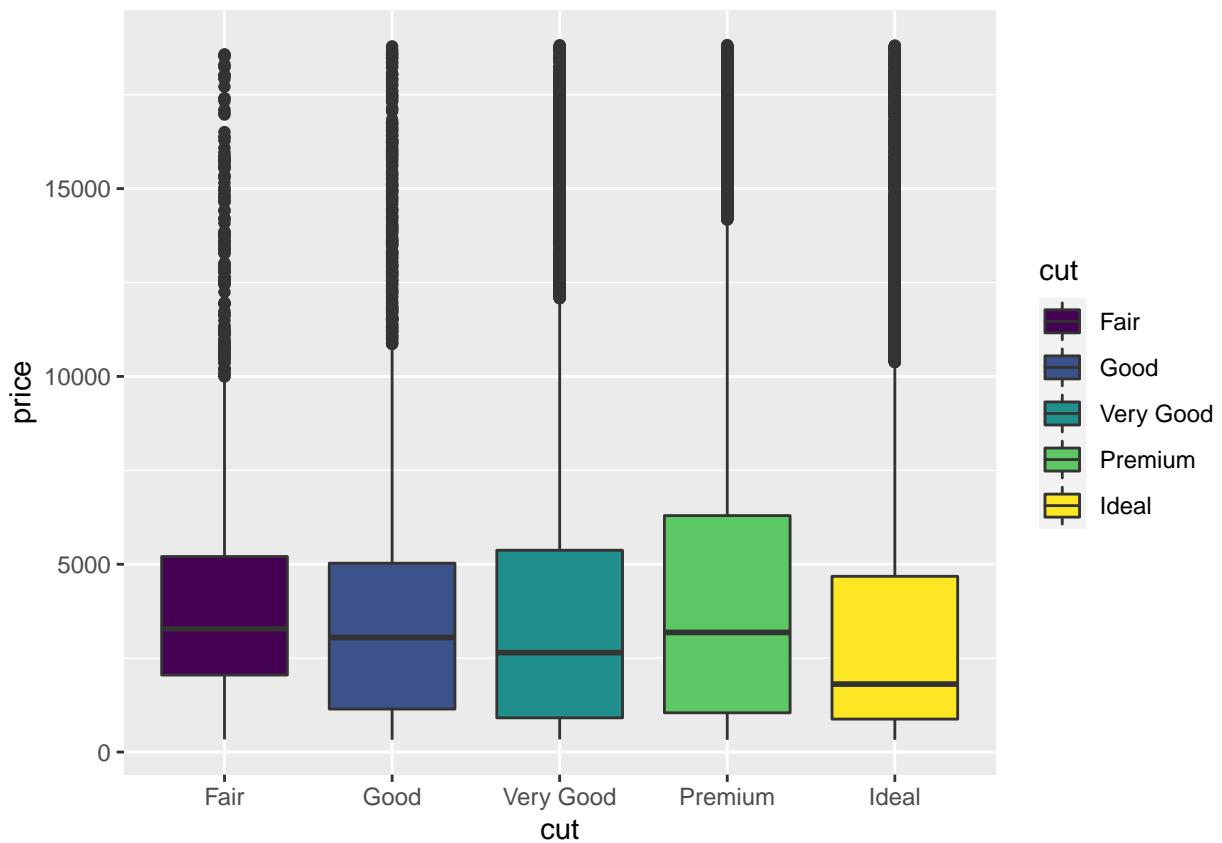
R: El peso del diamante influye fuertemente en el precio como lo podemos observar en el siguiente gráfico.

```
diamonds %>%
  ggplot(aes(carat, price)) +
  geom_hex(bins = 30)
```

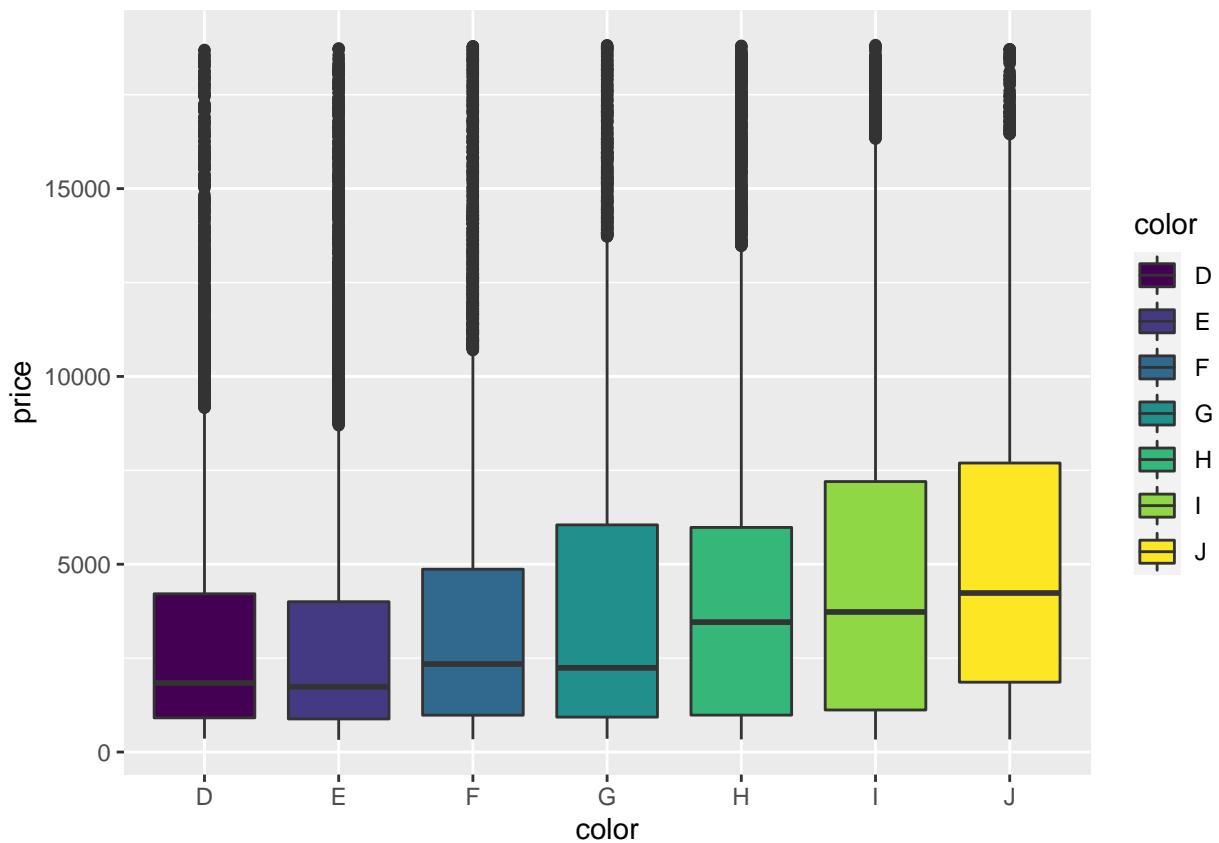


Por otro lado en las variables cut, color y clarity que se representan a través de boxplot no se visualiza una influencia más significativa en el precio como lo hace la variable carat.

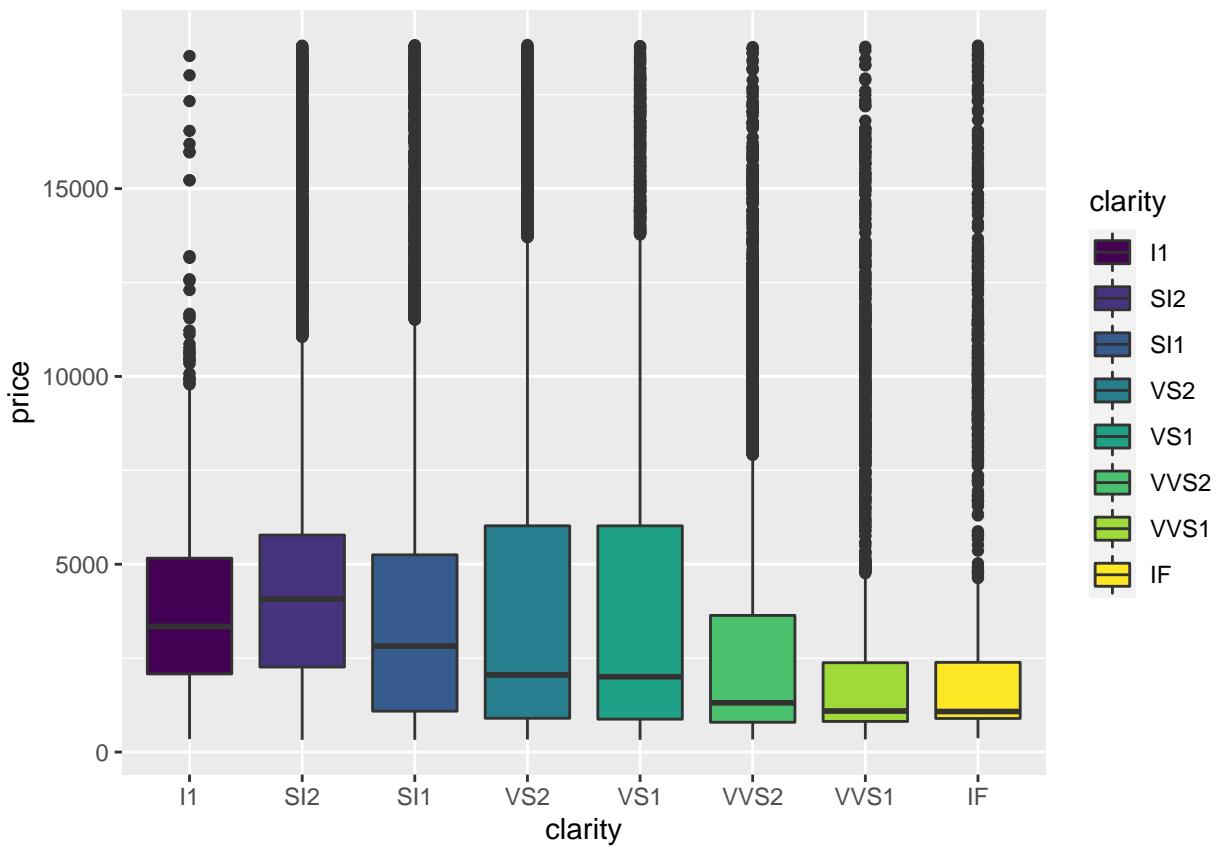
```
diamonds %>%
  ggplot(aes(cut, price, fill = cut)) +
  geom_boxplot()
```



```
diamonds %>%
  ggplot(aes(color, price, fill = color)) +
  geom_boxplot()
```

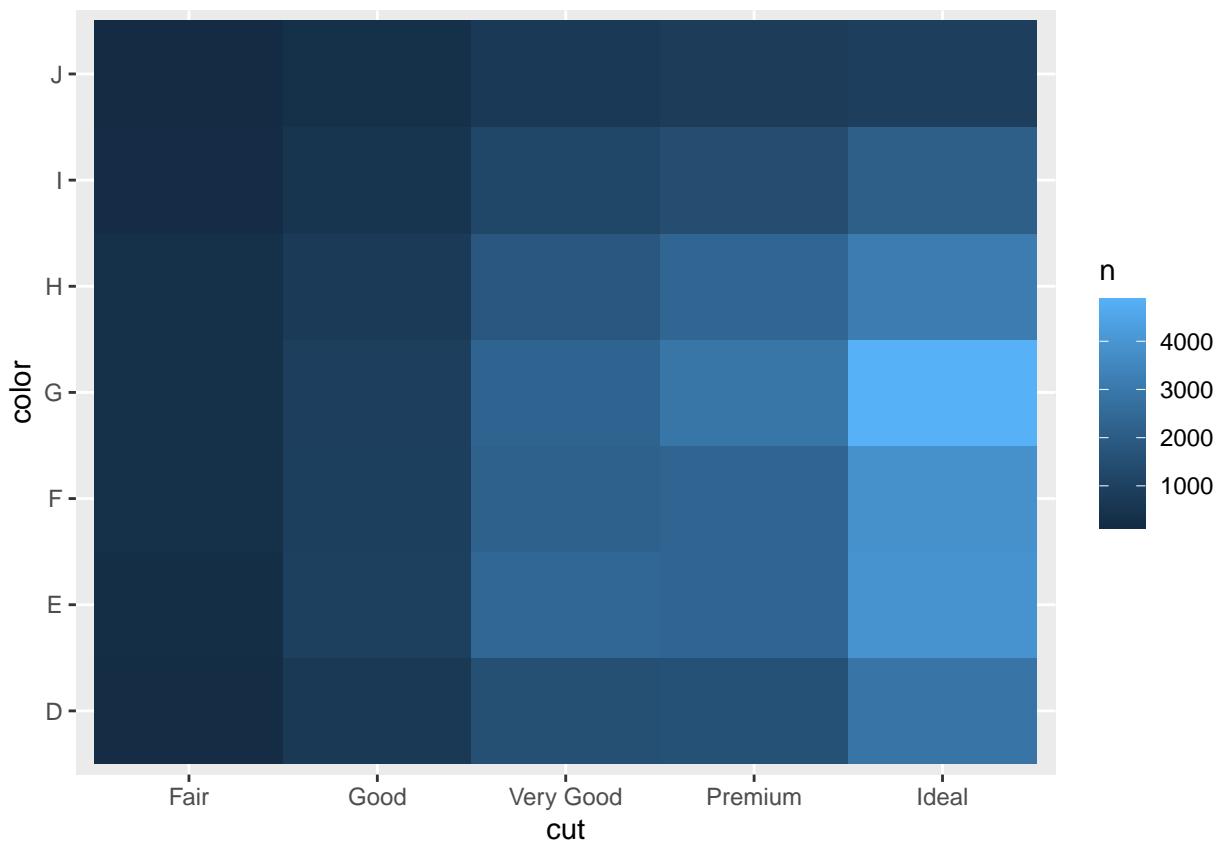


```
diamonds %>%
  ggplot(aes(clarity, price, fill = clarity)) +
  geom_boxplot()
```

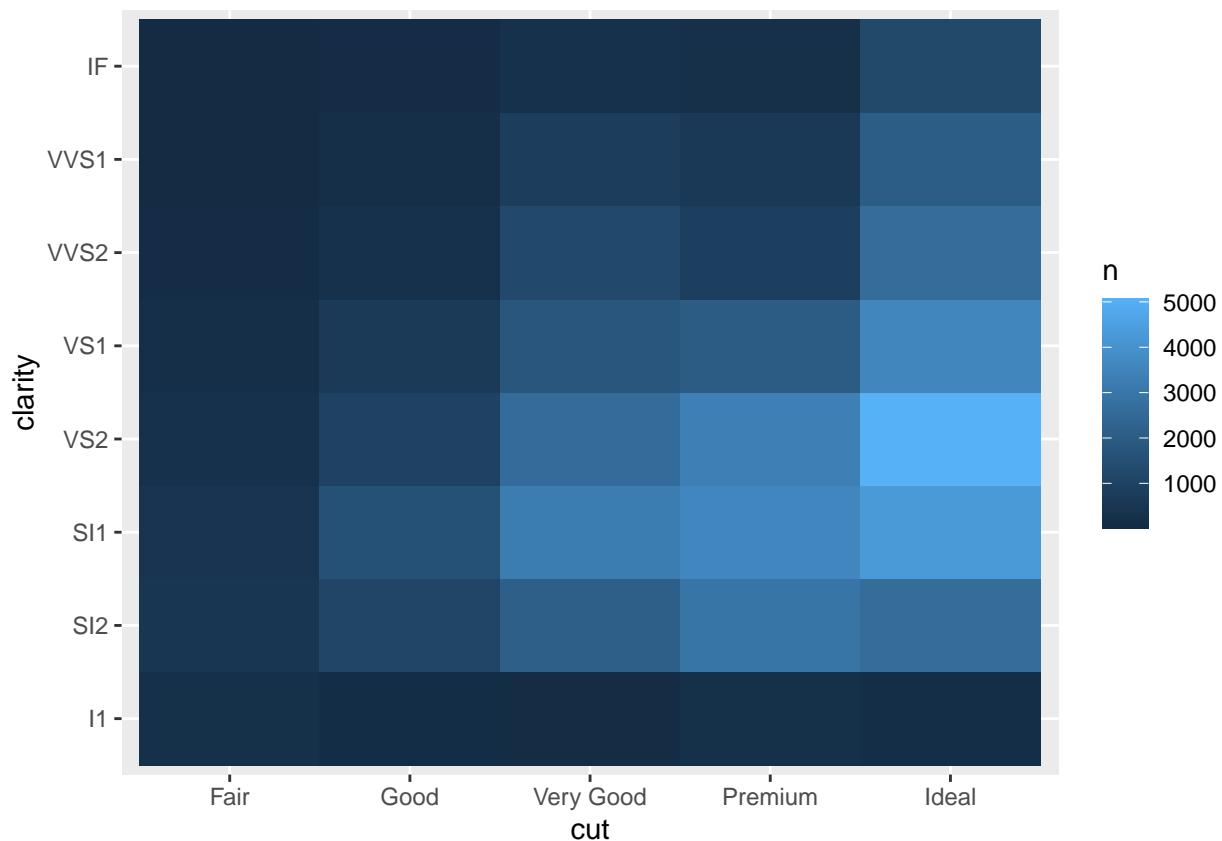


- ¿Qué variable del dataset de diamantes crees que es la que más correlacionada está con cut?

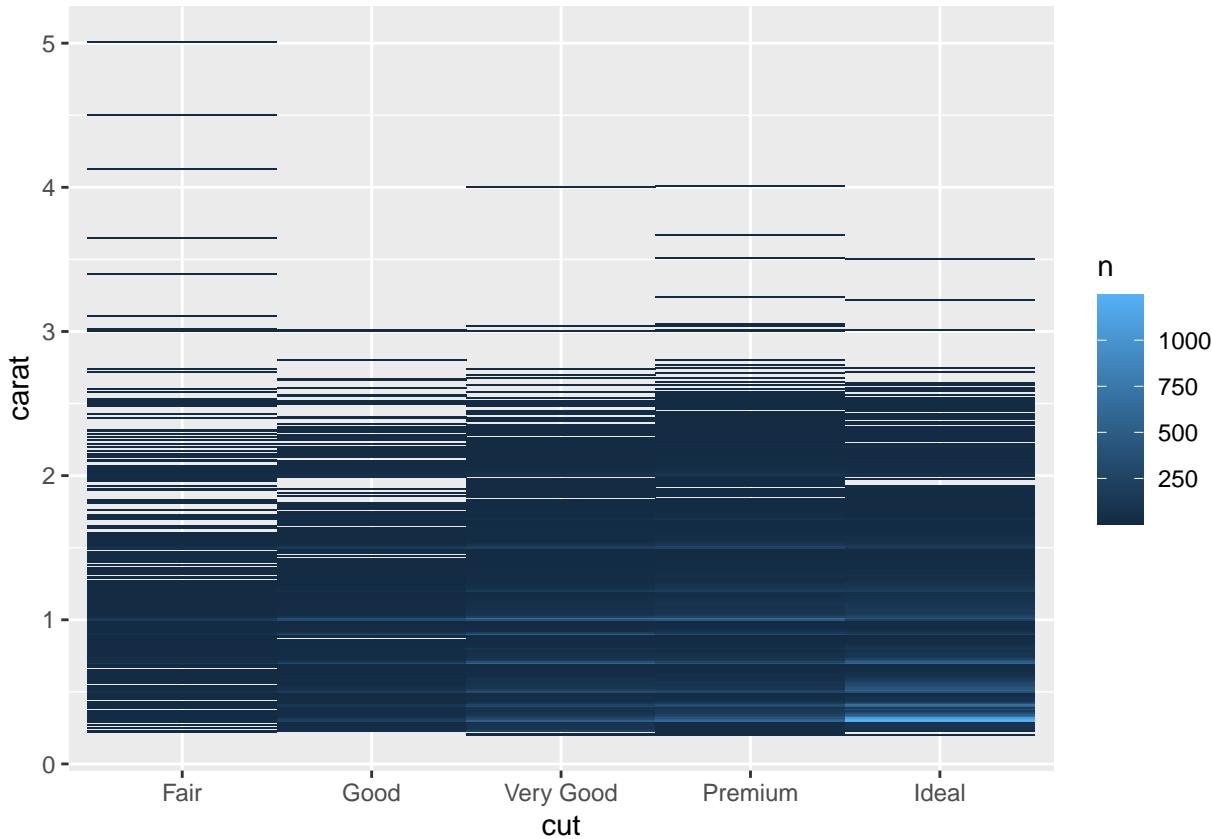
```
diamonds %>%
  count(color, cut) %>%
  ggplot(mapping = aes(x = cut, y = color)) +
  geom_tile(mapping = aes(fill = n))
```



```
# Al parecer donde hay mayor numero de ocurrencias es con clarity.
diamonds %>%
  count(clarity, cut) %>%
  ggplot(mapping = aes(x = cut, y = clarity)) +
  geom_tile(mapping = aes(fill = n))
```



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



- ¿Por qué combinar estas dos variables nos lleva a que los diamantes con peor calidad son los mas caros?

Pregunta 3

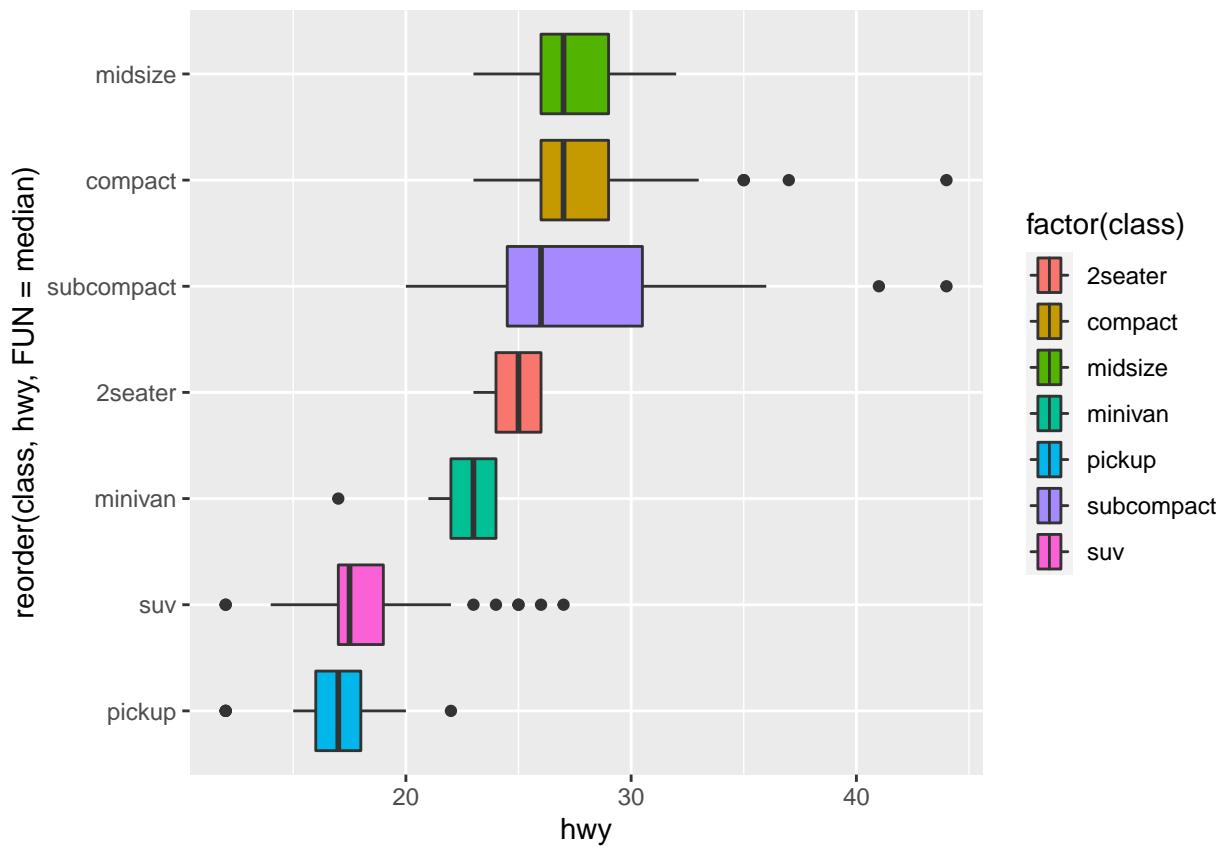
Instala el paquete de ggstance y úsalo para crear un boxplot horizontal. Compara el resultado con usar el coord_flip() que hemos visto en clase.

```
library(ggstance)

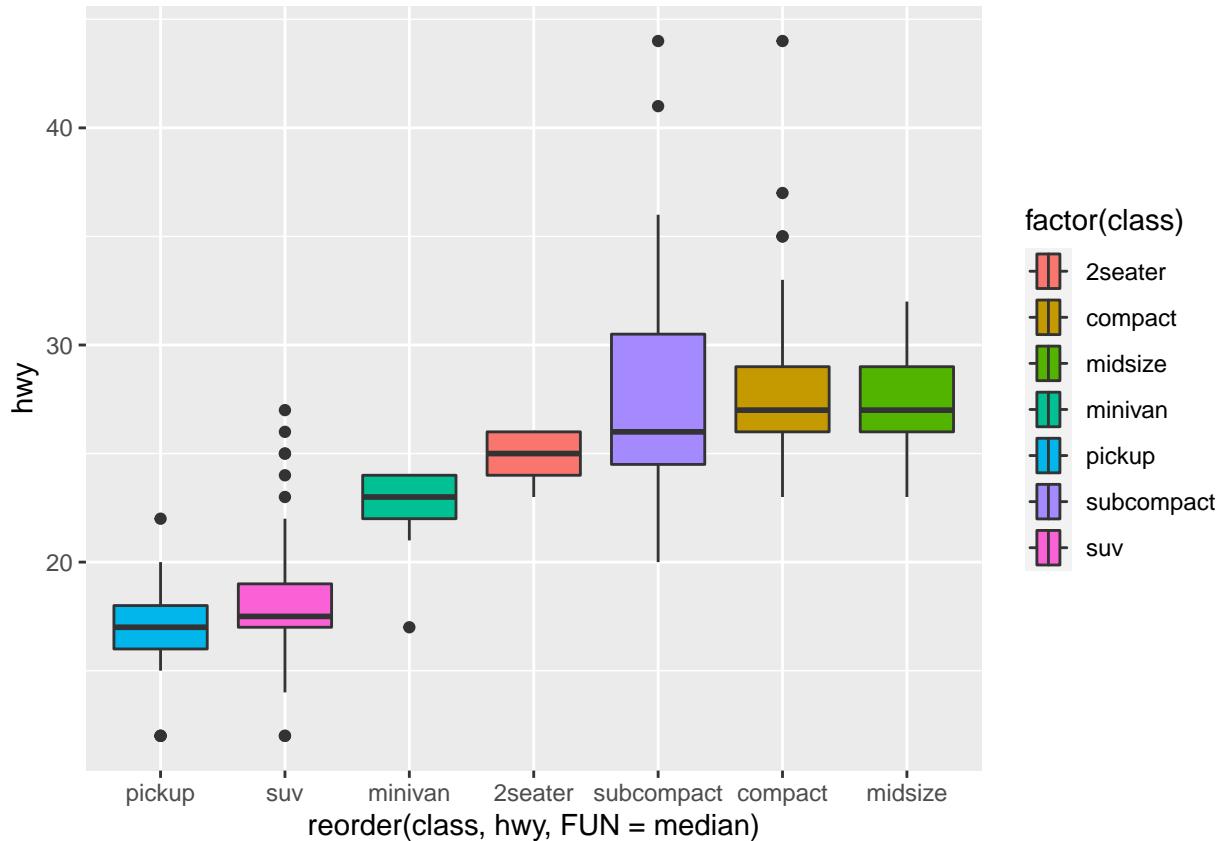
## 
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
## 
##     geom_errorbarh, GeomErrorbarh

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



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

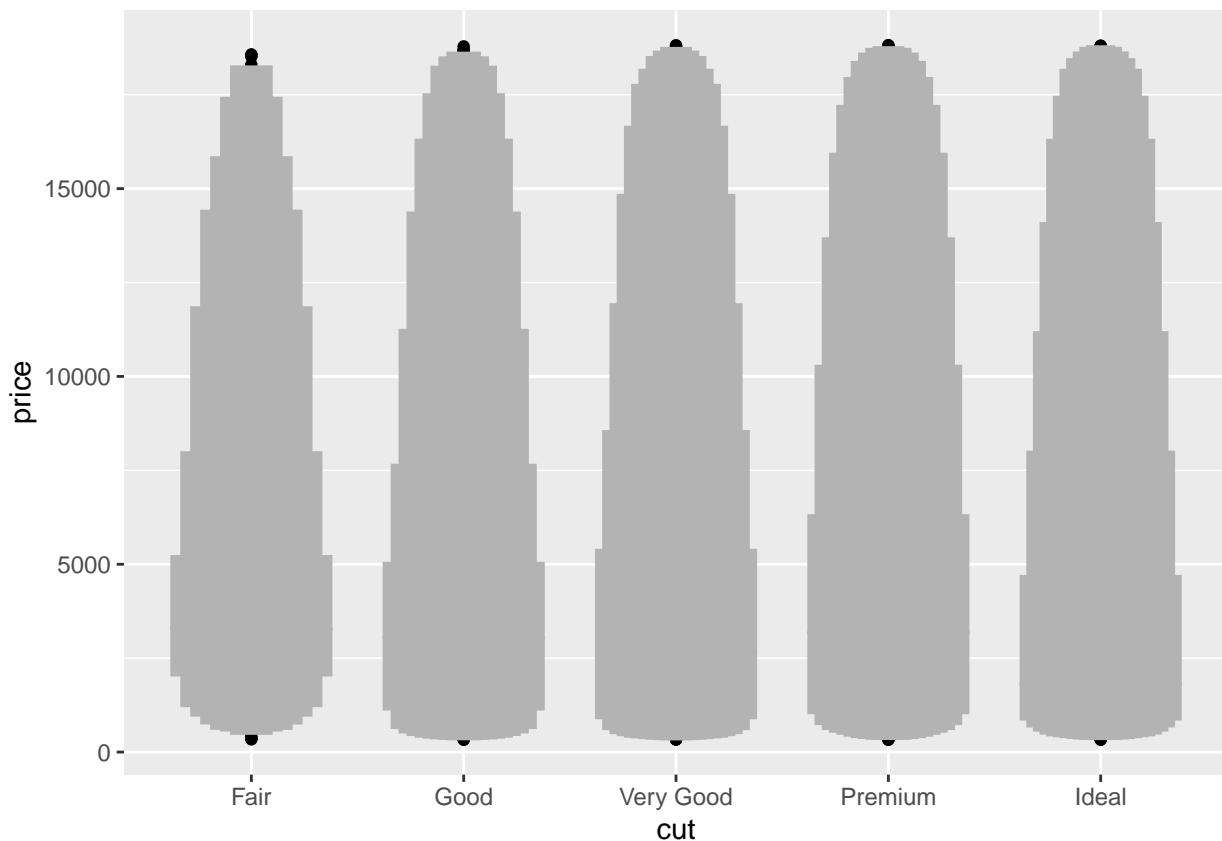


Pregunta 4

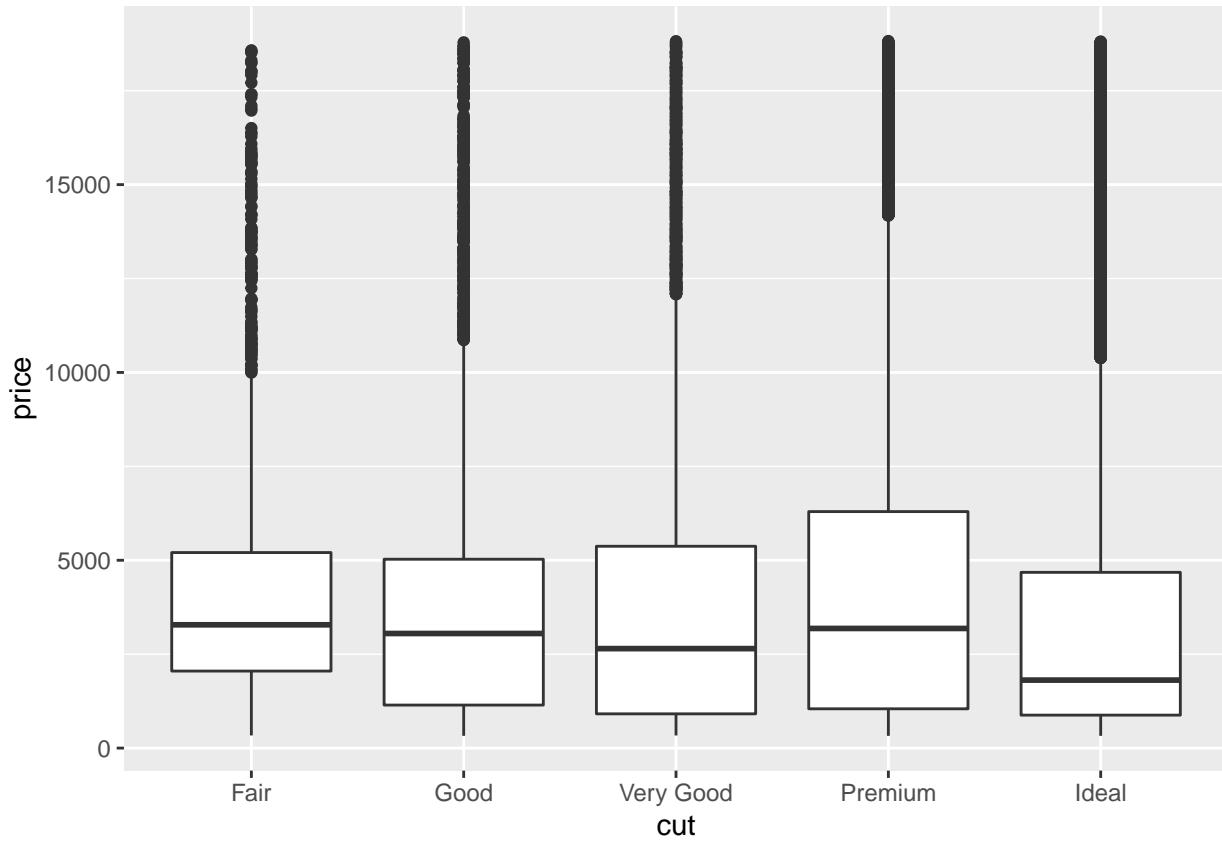
Los boxplots nacieron en una época donde los datasets eran mucho más pequeños y la palabra big data no era más que un concepto futurista. De ahí que los datos considerados con outliers tuvieran sentido que fueran representados con puntos dado que su existencia era más bien escasa o nula. Para solucionar este problema, existe el letter value plot del paquete lvplot. Instala dicho paquete y usa la geometría geom_lv() para mostrar la distribución de precio vs cut de los diamantes. ¿Qué observas y qué puedes interpretar a raíz de dicho gráfico?

```
library(lvplot)

diamonds %>%
  ggplot(aes(cut, price)) +
  geom_lv()
```



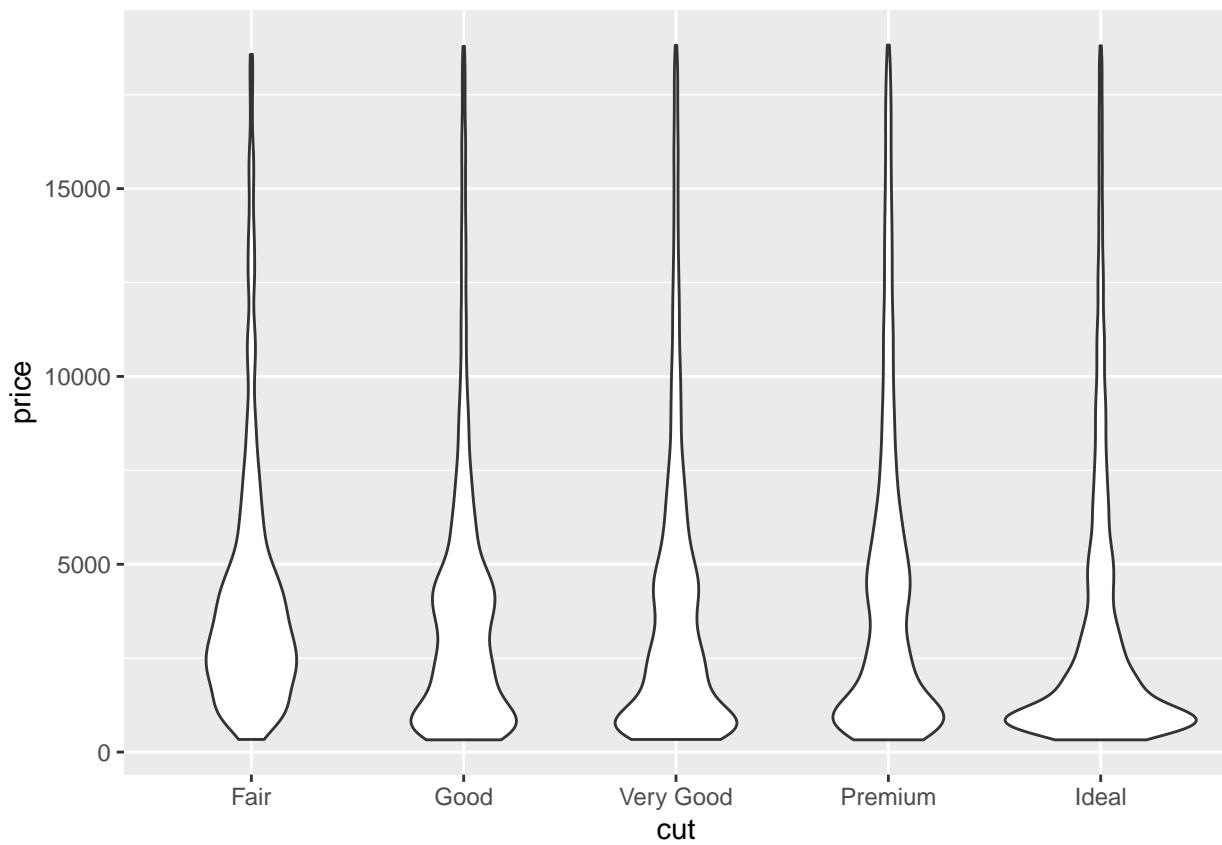
```
diamonds %>%
  ggplot(aes(cut, price)) +
  geom_boxplot()
```



Pregunta 5

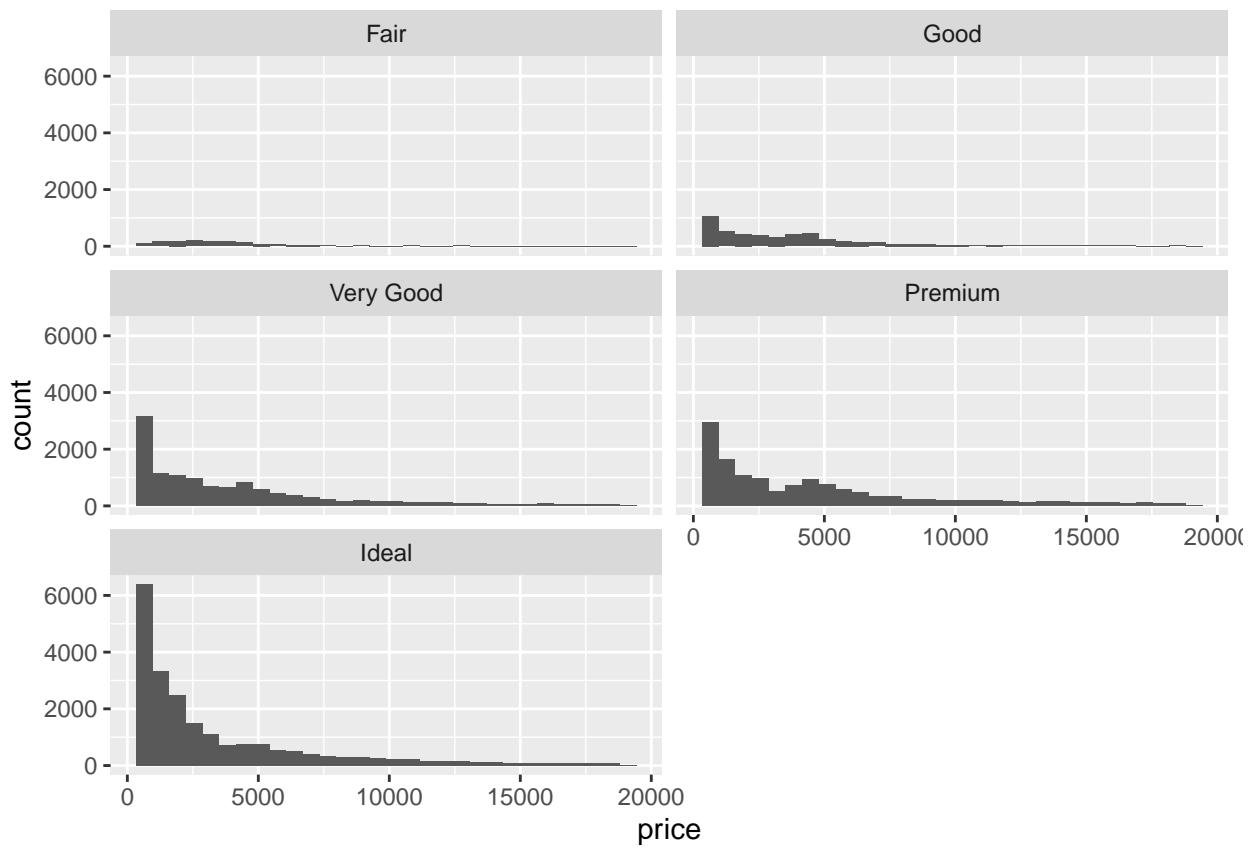
Compara el uso de la geometría geom_violin() con un facet de geom_histogram() y contra un geom_freqpoly() coloreado. Investiga cuales son los pros y los contras de cada uno de los tipos de representación.

```
diamonds %>%
  ggplot(aes(cut, price)) +
  geom_violin()
```

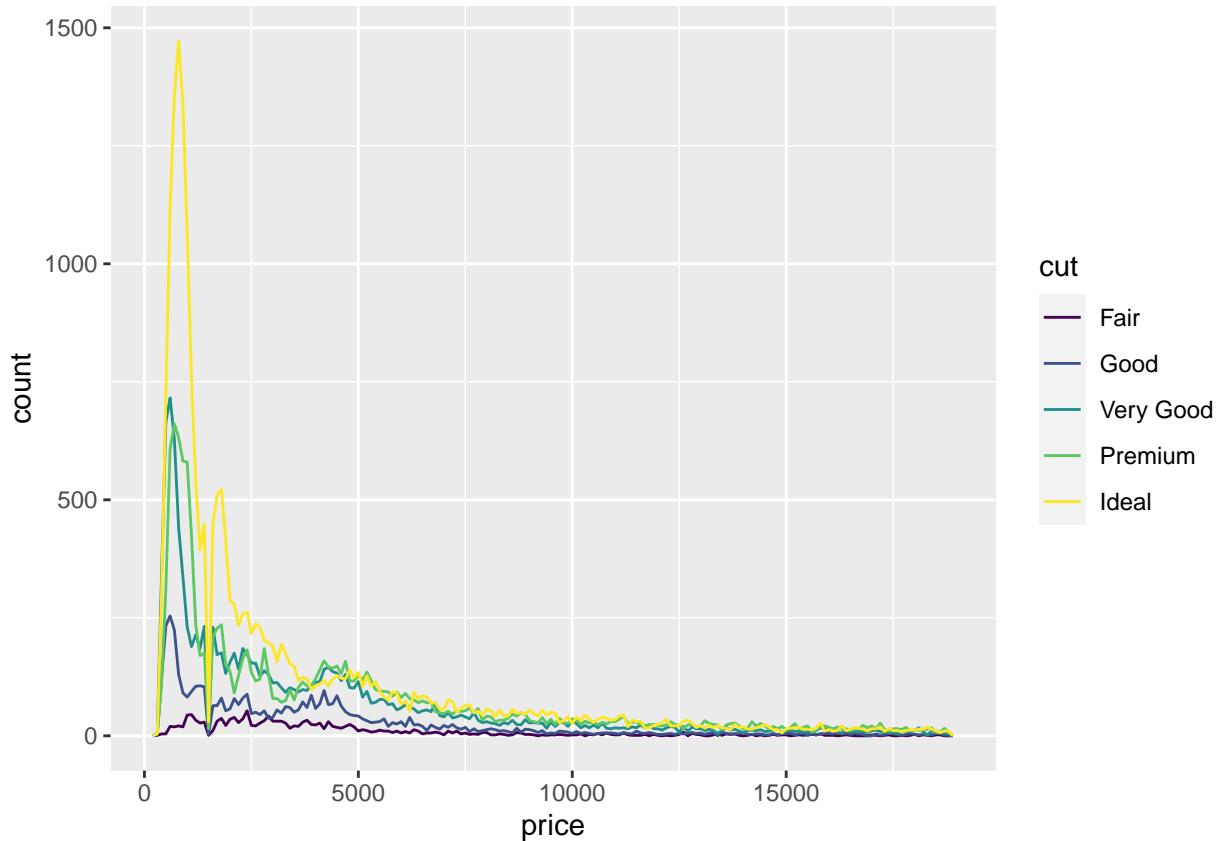


```
iamonds %>%
  ggplot(aes(price), binwidth = 100) +
  geom_histogram() +
  facet_wrap(~cut, nrow = 3)

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



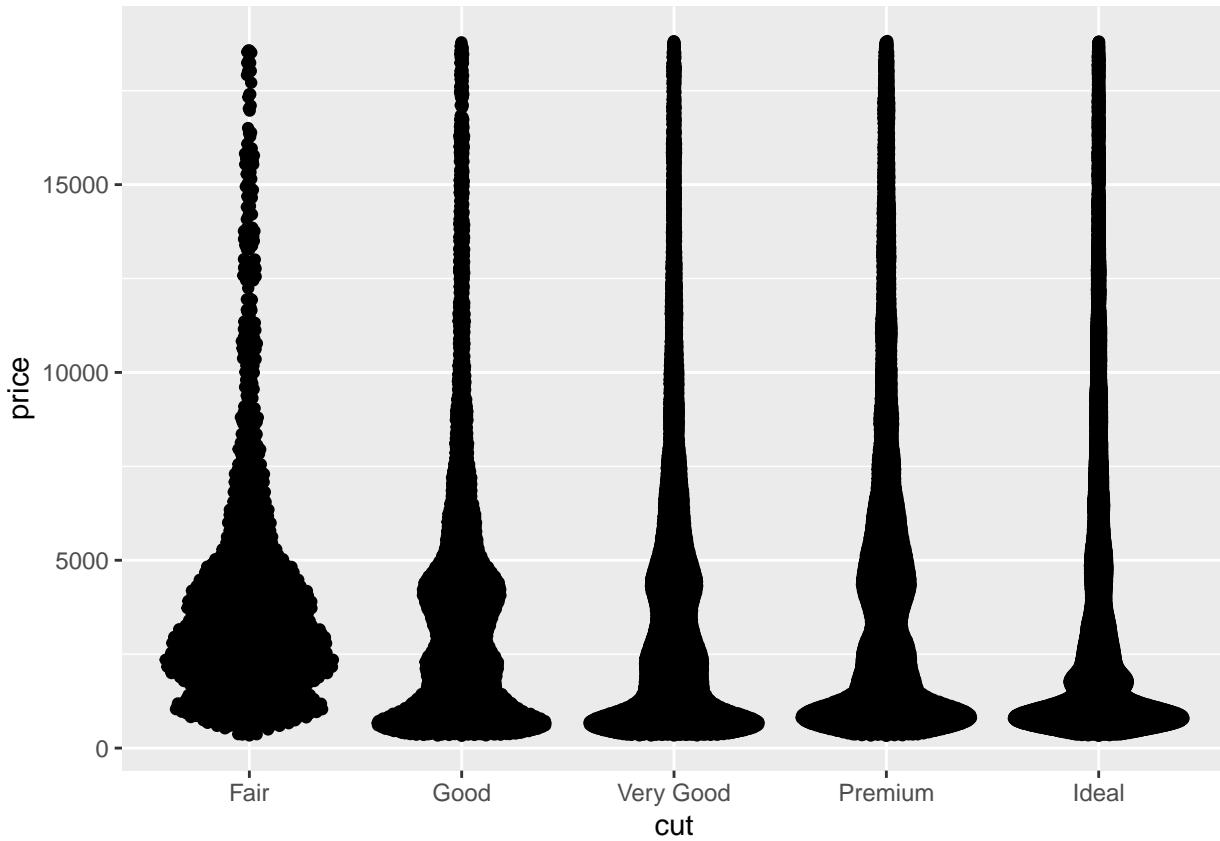
```
diamonds %>%
  ggplot(aes(price, color = cut)) +
  geom_freqpoly(binwidth = 100)
```



Pregunta 6

Si tenemos datasets pequeños, a veces es útil usar la opción que ya conocemos de `geom_jitter()` para ver la relación entre una variable continua y una variable categórica. El paquete de R `ggbeeswarm` tiene un par de métodos similares a `geom_jitter()` que te pueden ayudar a tal efecto. Lístalos y haz un gráfico con cada uno de ellos para ver qué descripción de los datos podemos extraer de cada uno. ¿A qué gráfico de los que ya has visto durante esta práctica se parece?

```
library(ggbeeswarm)
diamonds %>%
  ggplot(aes(cut, price)) +
  geom_quasirandom()
```



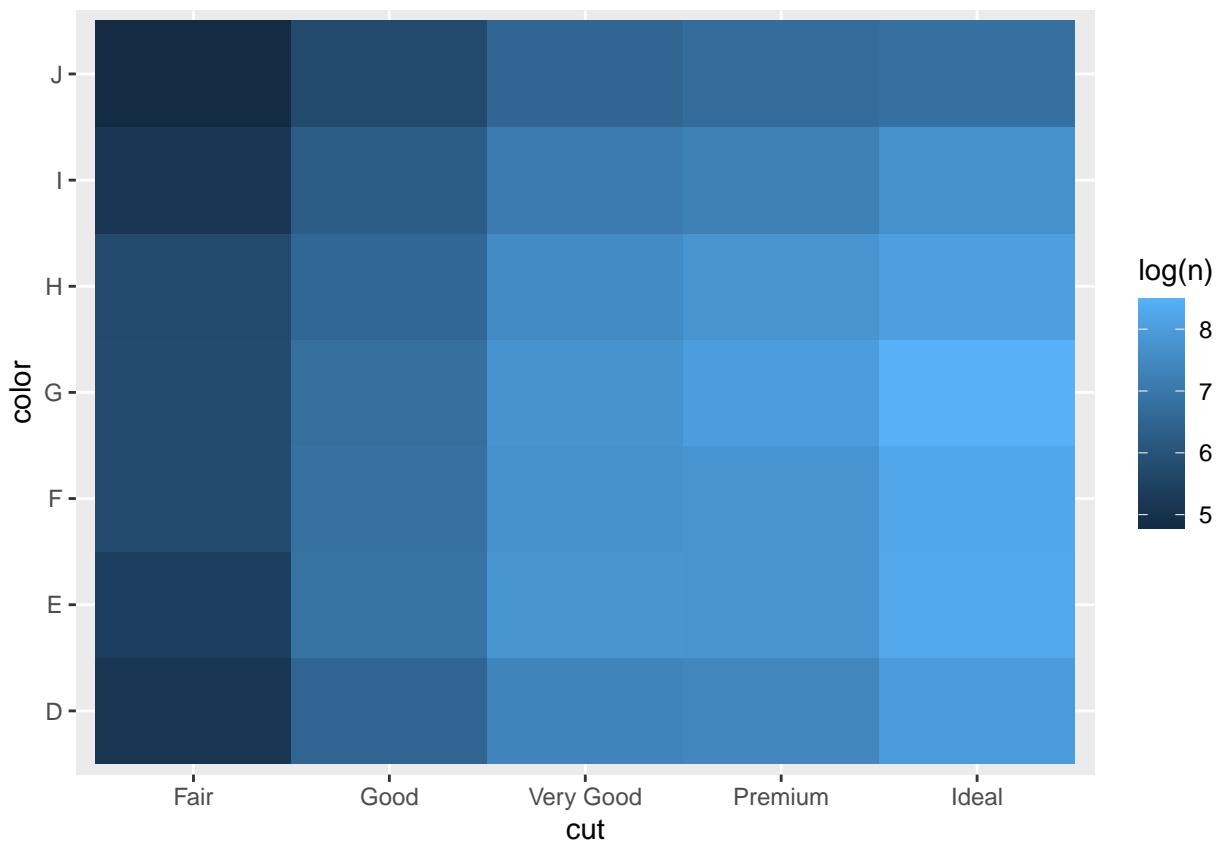
Pregunta 7

Los mapas de calor que hemos visto tienen un claro problema de elección de los colores.

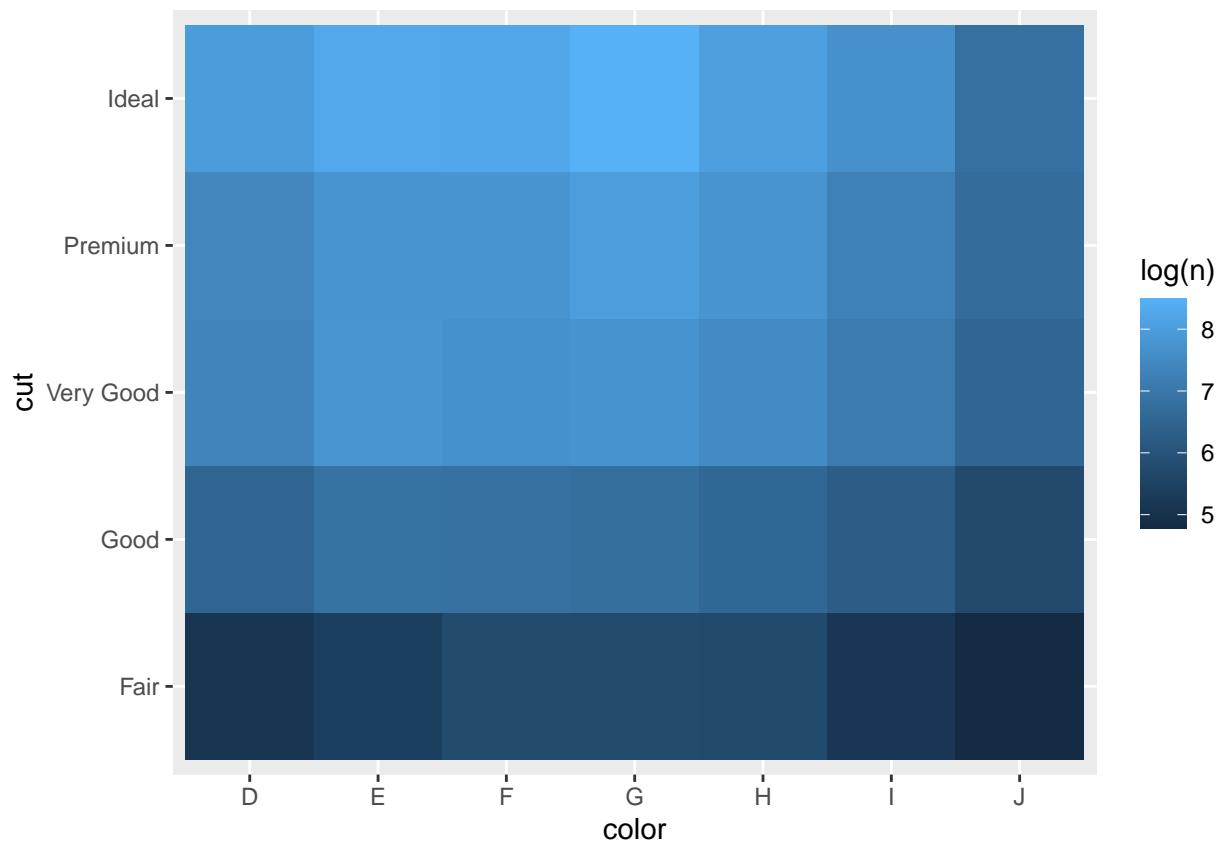
- ¿Cómo podríamos reescalar el campo count del dataset de diamantes cuando cruzamos color y cut para observar mejor la distribución de dicho cruce?
- ¿Por qué resulta mejor usar la estética aes(x = color, y = cut) en lugar de aes(x=cut, y = color)?

R: porque el ojo humano le resulta más fácil leer de izquierda a derecha.

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



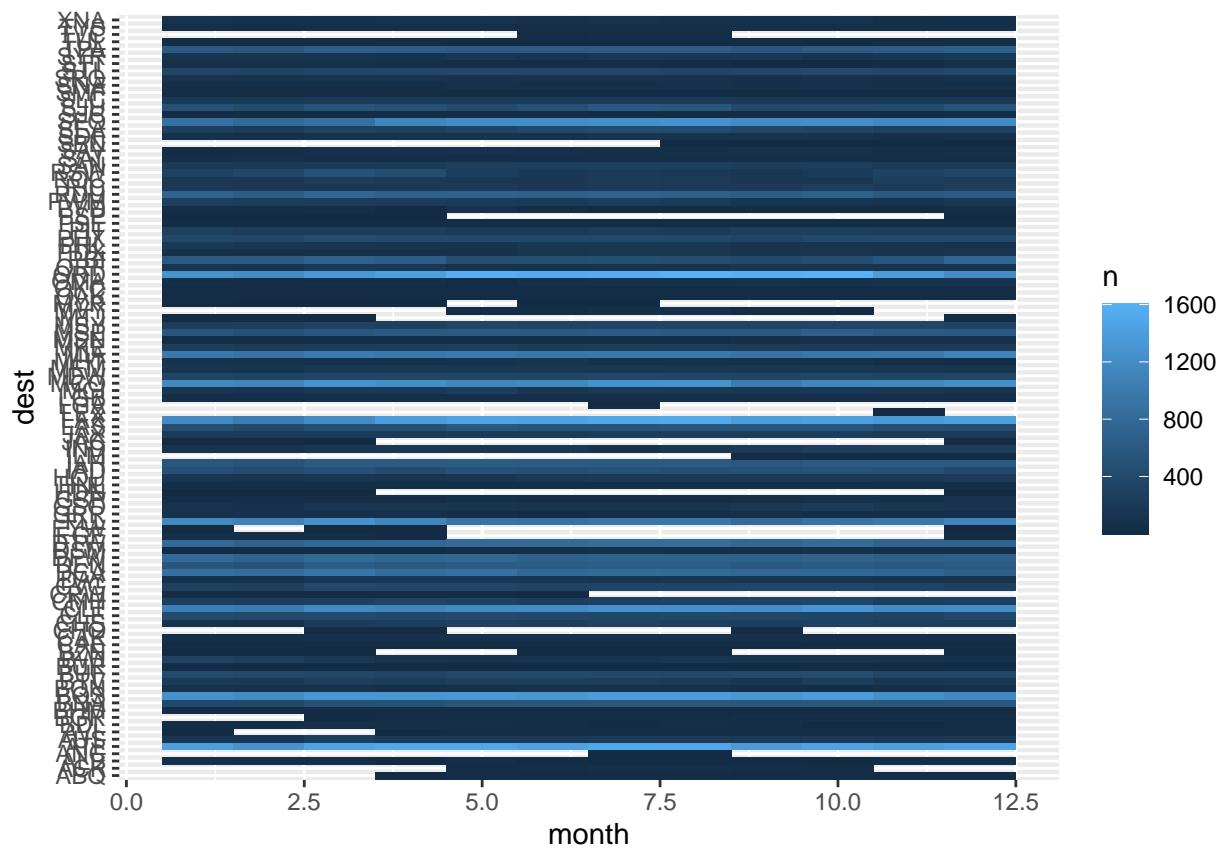
```
iamonds %>%
  count(color, cut) %>%
  ggplot(aes(color, cut))+
  geom_tile(aes(fill = log(n)))
```



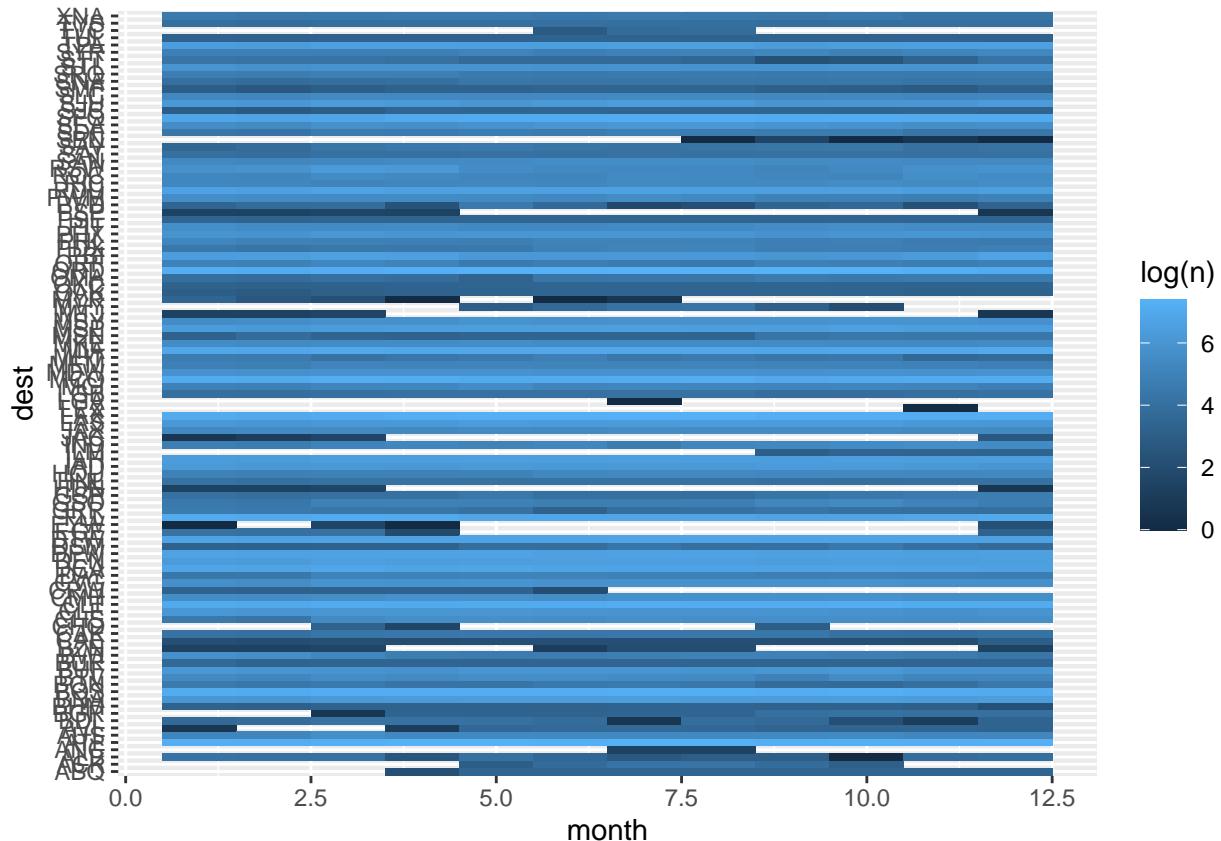
Pregunta 8

- Utiliza la geom_tile() junto con dplyr para explorar si el promedio del retraso de los vuelos varía con respecto al destino y mes del año.
- ¿Qué hace que este gráfico sea difícil de leer o de interpretar?
- ¿Cómo puedes mejorar la visualización?

```
flights %>%
  count(month, dest) %>%
  ggplot(aes(month, dest)) +
  geom_tile(aes(fill = n))
```



```
#Aplicando fill = log(n) se mejora un poco la visualizacion
flights %>%
  count(month, dest) %>%
  ggplot(aes(month, dest)) +
  geom_tile(aes(fill = log(n)))
```



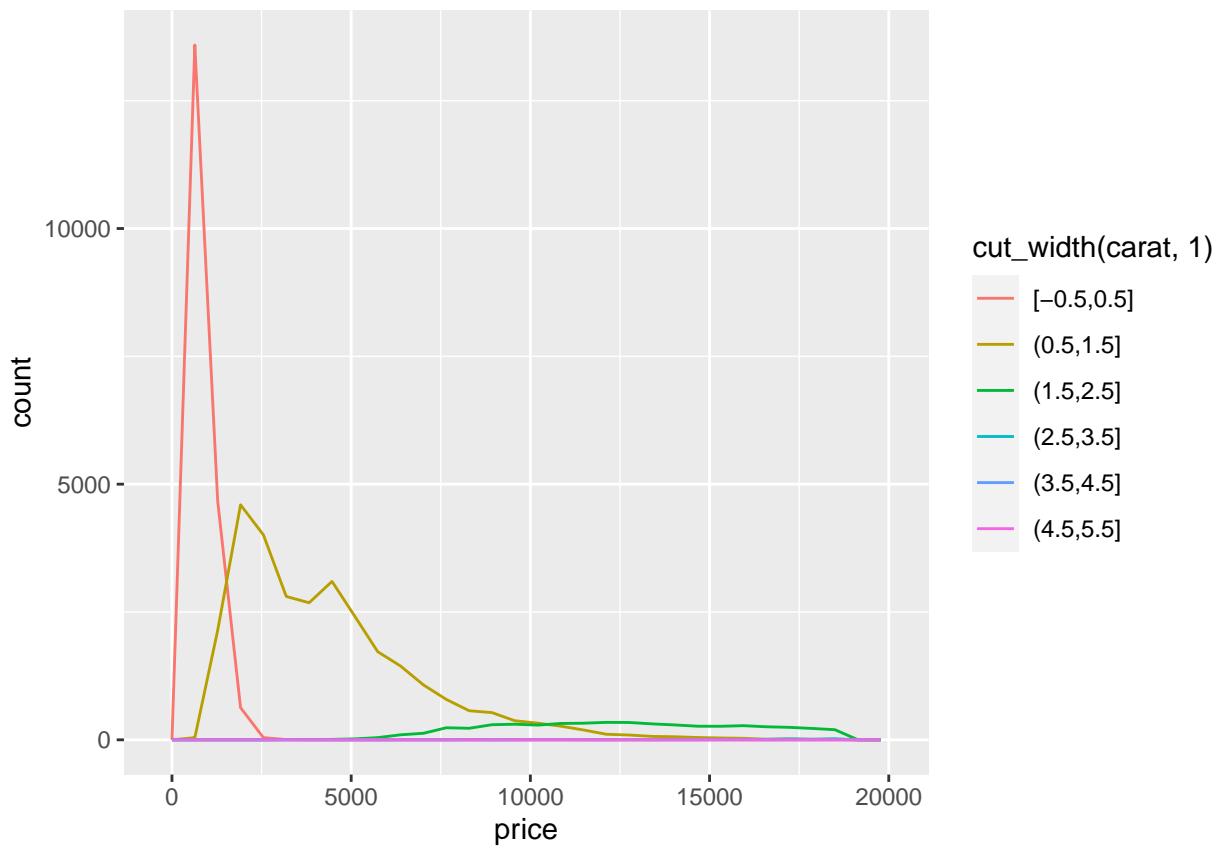
Pregunta 9

En lugar de hacer un resumen de la distribución condicional de dos variables numéricas con un boxplot, se puede usar un polígono de frecuencias.

- ¿Qué hay que tener en cuenta cuando usas `cut_width()` o cuando usas `cut_number()`?
- ¿Cómo influye este hecho en la visualización 2D de carat y price?
- Da la mejor visualización posible de carat dividido por price.

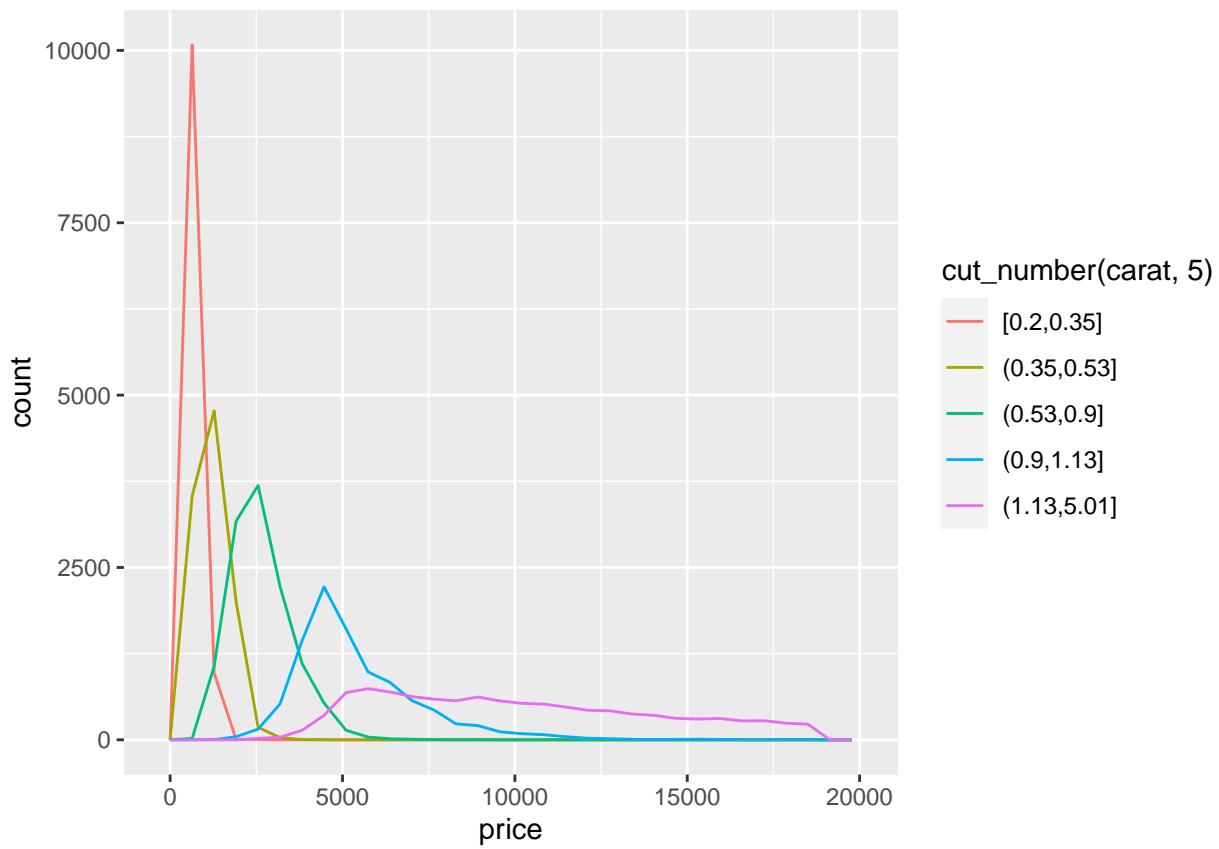
```
ggplot(diamonds,aes(price, colour =cut_width(carat, 1.0)))+
  geom_freqpoly()

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



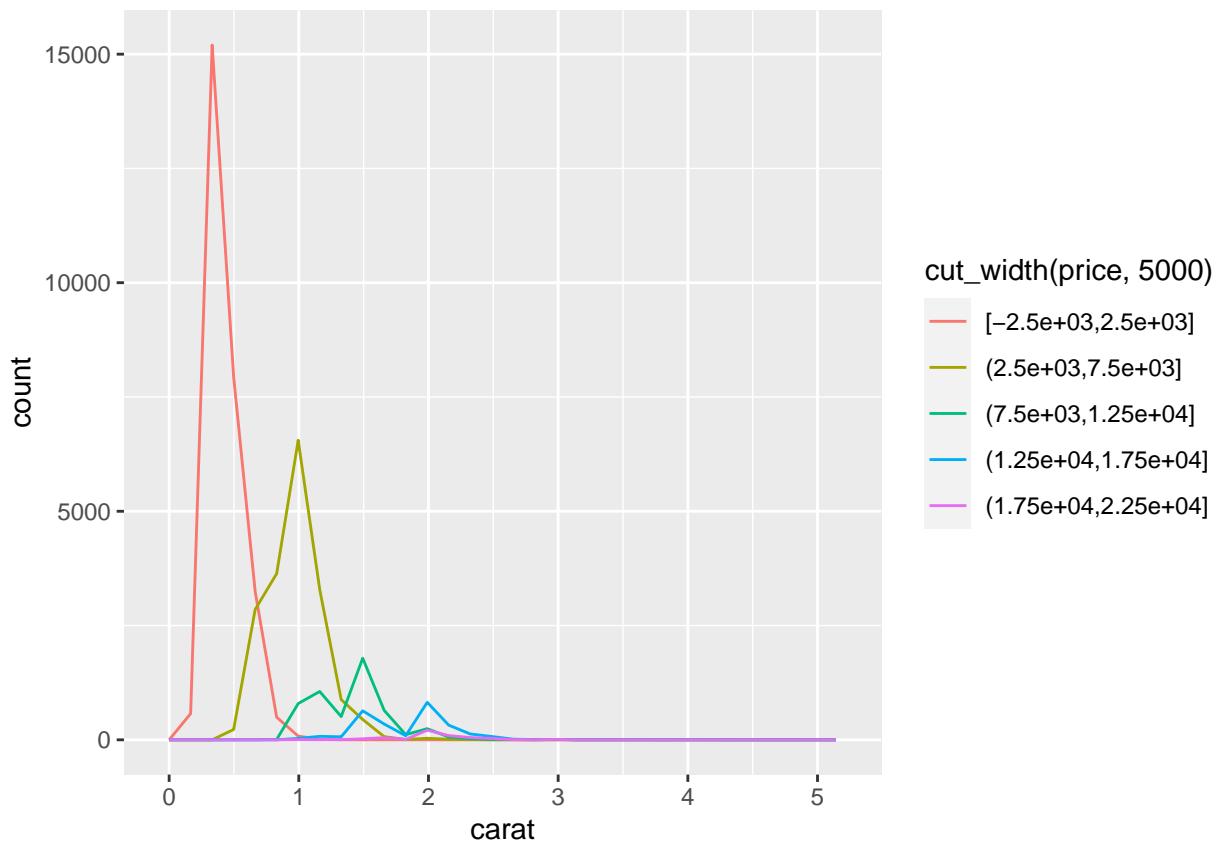
```
ggplot(diamonds,aes(price, colour =cut_number(carat, 5)))+
  geom_freqpoly()
```

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



```
ggplot(diamonds,aes(carat, colour =cut_width(price, 5000)))+
  geom_freqpoly()
```

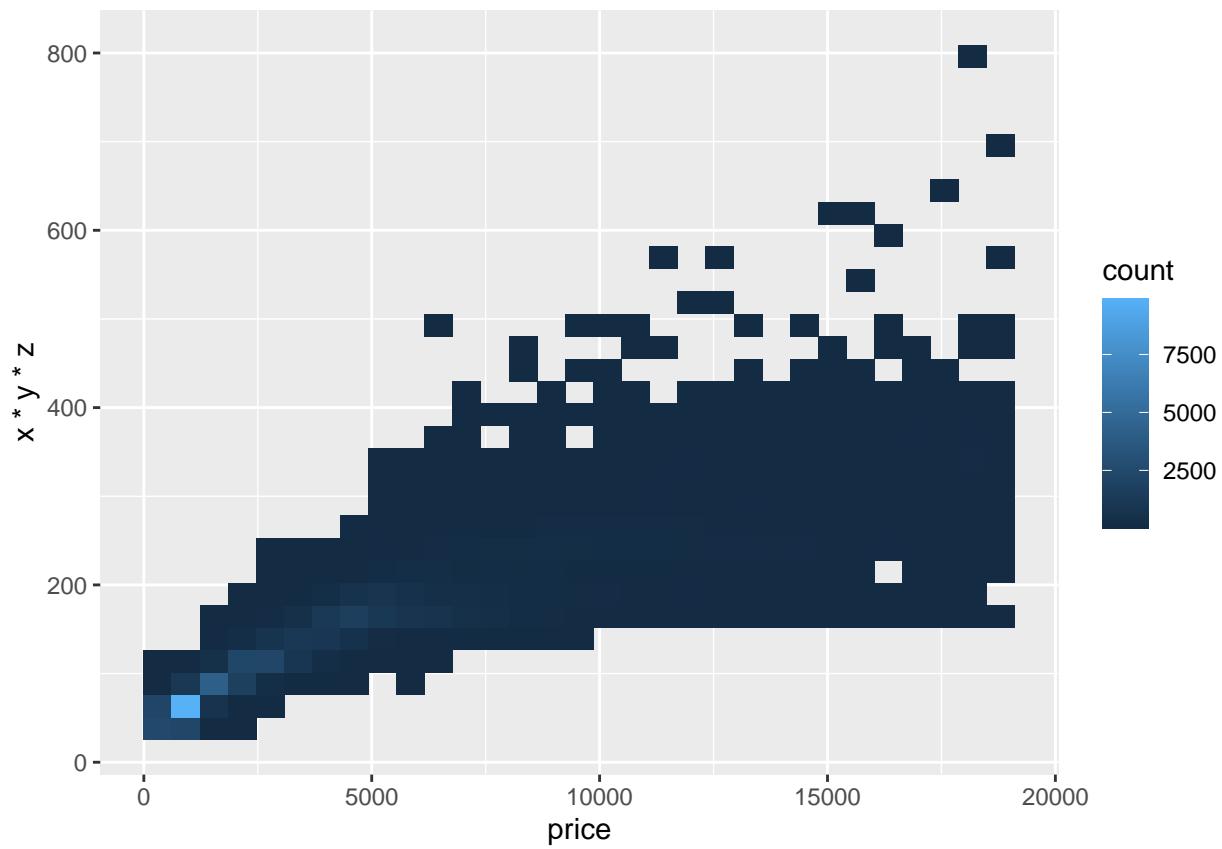
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Pregunta 10

Compara la distribución del precio de los diamantes grandes vs diamantes pequeños. Elige el concepto de grande y pequeño que consideres. Comenta el resultado.

```
diamonds %>%
  filter(between(x, 2, 20)) %>%
  filter(between(y, 2, 20)) %>%
  filter(between(z, 2, 20)) %>%
  ggplot(aes(price, x*y*z)) +
  geom_bin2d()
```



Pregunta 11

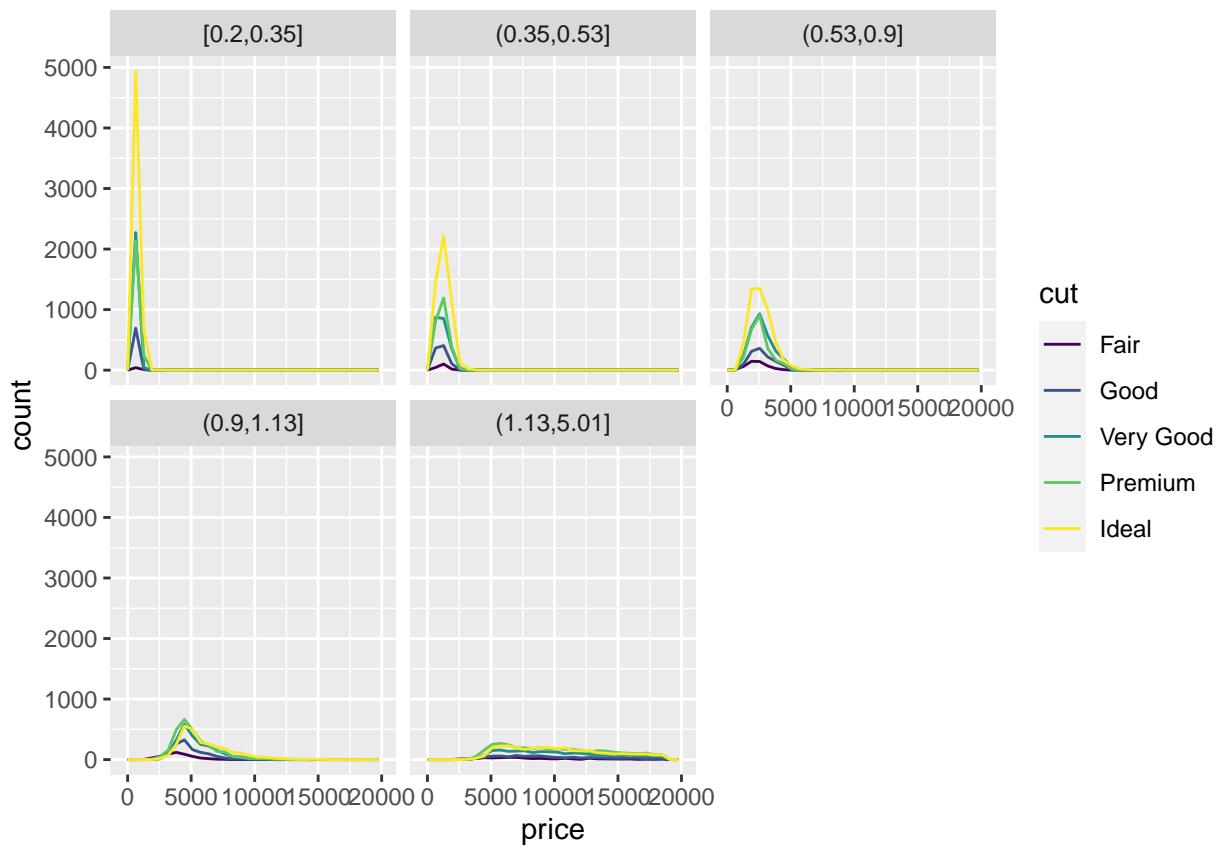
Combina diferentes técnicas de ggplot para visualizar la distribución combinada de cut, carat y precio.

```

diamonds %>%
  ggplot(aes(price, colour = cut)) +
  geom_freqpoly() +
  facet_wrap(~cut_number(carat), nrow = 2)

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

```

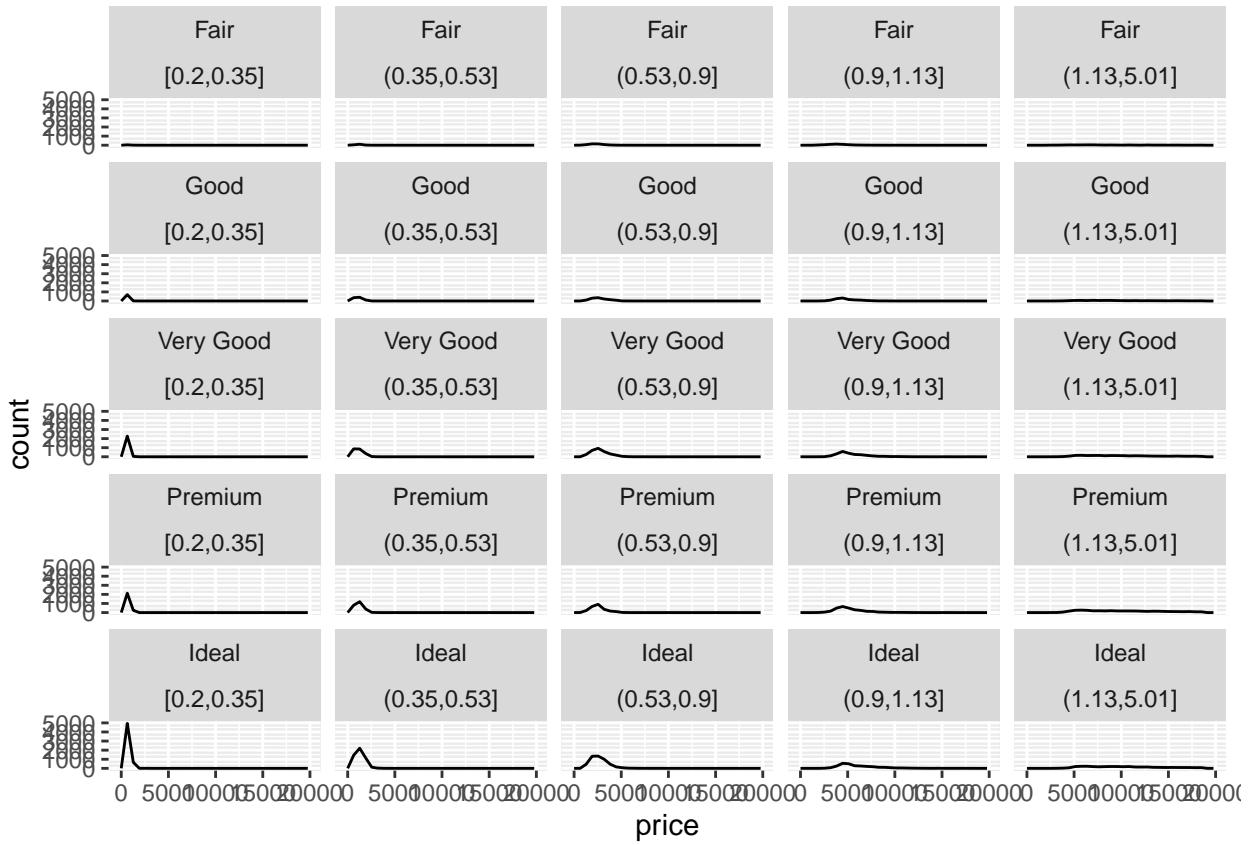


```

diamonds %>%
  ggplot(aes(price)) +
  geom_freqpoly() +
  facet_wrap(cut ~ cut_number(carat, 5))

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

```



Pregunta 12

Los plots en 2D pueden revelar outliers que no se ven en plots de una sola dimensión. Por ejemplo, algunos puntos del plot dado por hacen destacar muchísimo los outliers combinando x con y, a pesar de que por separado parecen valores normales.

Intenta averiguar por qué un scatterplot resulta más efectivo en este caso que un gráfico con agrupaciones.

```
ggplot(data = diamonds) + geom_point(mapping = aes(x = x, y = y)) + coord_cartesian(xlim = c(4,12), ylim = c(4,12))
```

```
diamonds %>%
  ggplot() +
  geom_point(aes(x,y)) +
  coord_cartesian(xlim = c(4,12), ylim = c(4,12))
```

