

Tarea 2

Master en Big Data. Fundamentos Matemáticos del Análisis de Datos (FMAD).

Nicolás Núñez de Cela Román

Curso 2021-22. Última actualización: 2021-09-24

Preliminares

- Comenzamos con las librerías que vamos a necesitar durante la tarea.

```
library(tidyverse)
library(dplyr, warn.conflicts = FALSE)
options(dplyr.summarise.inform = FALSE)
```

Ejercicio 1. Simulando variables aleatorias discretas.

Apartado 1: La variable aleatoria discreta X_1 tiene esta tabla de densidad de probabilidad (es la variable que se usa como ejemplo en la Sesión):

valor de X_1	0	1	2	3
Probabilidad de ese valor $P(X = x_i)$	$\frac{64}{125}$	$\frac{48}{125}$	$\frac{12}{125}$	$\frac{1}{125}$

Calcula la media y la varianza teóricas de esta variable.

Para calcular la media teórica, calculamos un vector con los valores y otro con sus probabilidades. Con ellos, calculamos su producto escalar para calcular la media teórica, dada por $\mu = \sum_{i=0}^3 x_i p_i$. Para calcular la varianza, simplemente hacemos uso de la definición:

$$\sigma^2 = \sum_{i=0}^3 (x_i - \mu)^2 p_i$$

```
valor <- seq(0,3,by = 1)
prob <- c(64,48,12,1)
prob <- prob/125

(media_teorica <- valor%*%prob %>%
  .[1,1])

## [1] 0.6
```

```
(varianza_teorica <- sum((valor-media_teorica)^2*prob))
```

```
## [1] 0.48
```

Apartado 2: Combina `sample` con `replicate` para simular cien mil muestras de tamaño 10 de esta variable X_1 . Estudia la distribución de las medias muestrales como hemos hecho en ejemplos previos, ilustrando con gráficas la distribución de esas medias muestrales. Cambia después el tamaño de la muestra a 30 y repite el análisis.

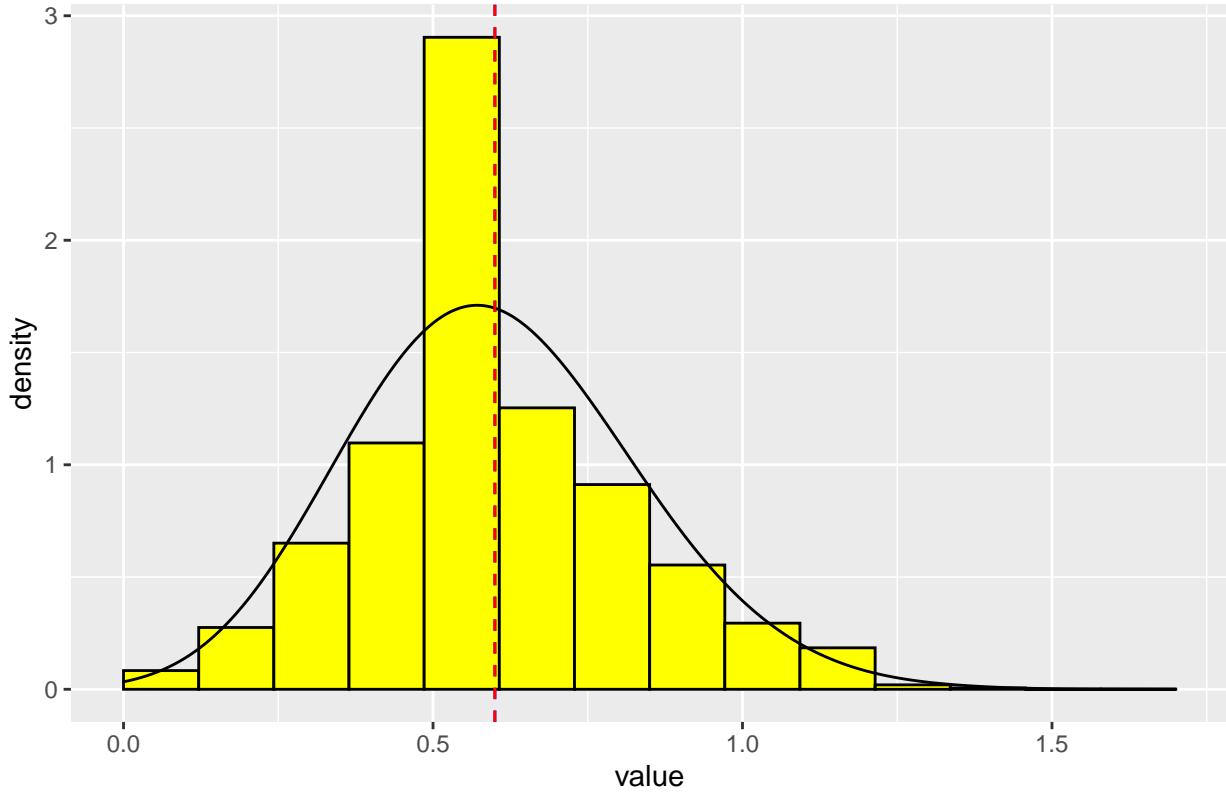
Definimos el número de veces que vamos a replicar un código, k y el tamaño del sample, n . Luego utilizamos el comando `replicate`, combinado con `sample`, donde calculamos muestras con los posibles valores de x_1 y con las probabilidades de dichos valores.

```
k = 100000  
n = 10  
  
muestras1 <- replicate(k, {  
  x1 = sample(valor, n ,replace = TRUE, prob)  
  mean(x1)  
})
```

Para el análisis de dichas muestras, representamos el histograma y la línea de densidad, junto con la media de las muestras y la media teórica de la población.

```
cortes = seq(min(muestras1),max(muestras1),length.out = 15)  
  
muestras1 %>%  
  as_tibble() %>%  
  ggplot() +  
    geom_histogram(mapping = aes(x = value, y = stat(density)),breaks = cortes, color ="black", fill = "#  
    geom_density(mapping = aes(x = value), adjust = 4) +  
    geom_vline(xintercept = media_teorica, color = "blue", linetype = "dashed") +  
    geom_vline(xintercept = mean(muestras1), color = "red", linetype = "dashed") +  
    ggtitle("Distribución de medias muestrales n = 10")
```

Distribución de medias muestrales $n = 10$



Vemos que no se observa la media de la población porque está superpuesta con la media del conjunto de muestras. Esto sucede en general, pero dependiendo del conjunto de muestras que nos toque, la coincidencia será mayor o menor.

Vamos a afianzar este pensamiento calculando muestras más grandes. Repetimos el código con 30 muestras.

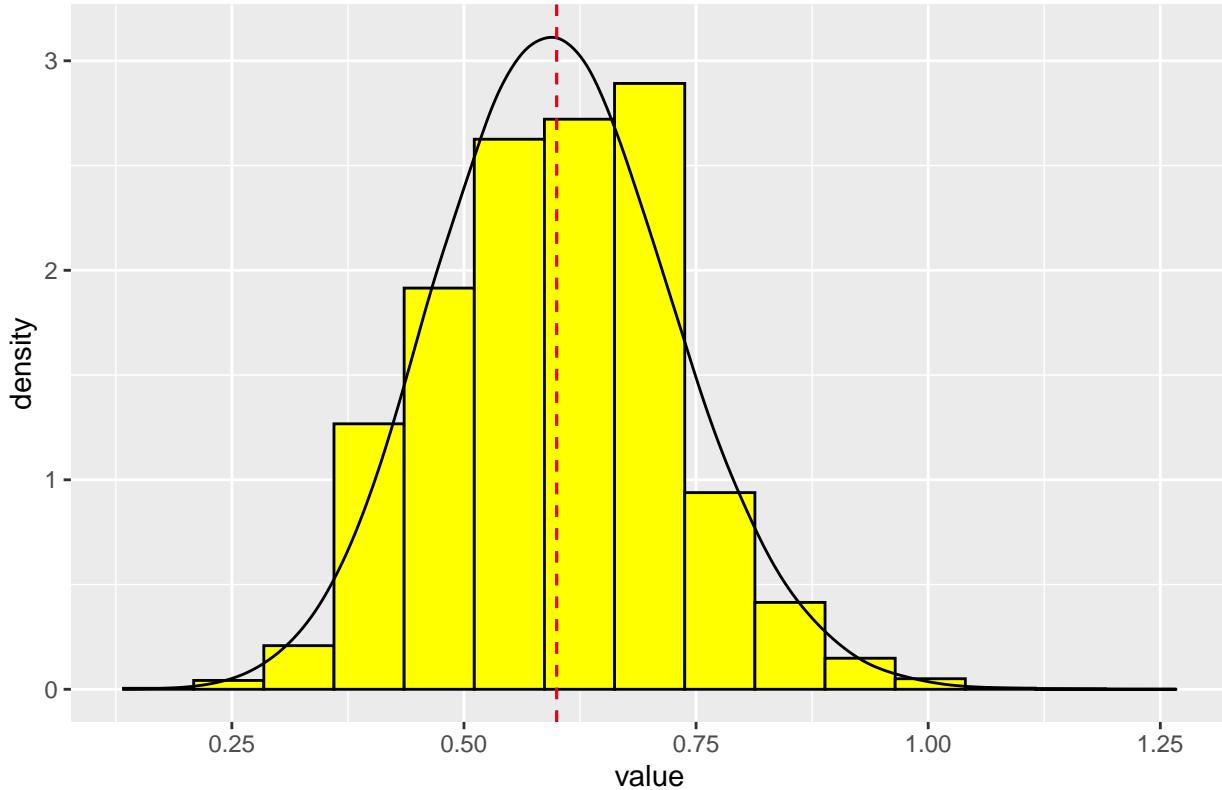
```
k = 100000
n = 30

muestras2 <- replicate(k, {
  x1 = sample(valor, n, replace = TRUE, prob)
  mean(x1)
})

cortes2 = seq(min(muestras2), max(muestras2), length.out = 16)

muestras2 %>%
  as_tibble() %>%
  ggplot(main = ) +
  geom_histogram(mapping = aes(x = value, y = stat(density)), breaks = cortes2, color ="black", fill = "yellow")
  geom_density(mapping = aes(x = value), adjust = 2) +
  geom_vline(xintercept = media_teorica, color = "blue", linetype = "dashed") +
  geom_vline(xintercept = mean(muestras2), color = "red", linetype = "dashed") +
  ggtitle("Distribución de medias muestrales n = 30")
```

Distribución de medias muestrales $n = 30$



Vemos como se asemeja cada vez más a una distribución normal, tal y como debe ser. Además, la media de las medias de las muestras coincide con la media de la población.

Apartado 3: La variable aleatoria discreta X_2 tiene esta tabla de densidad de probabilidad:

valor de X_2	0	1	2
Probabilidad de ese valor $P(X = x_i)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$

Suponemos que X_1 y X_2 son independientes. ¿Qué valores puede tomar la suma $X_1 + X_2$? ¿Cuál es su tabla de probabilidad?

```
valor_2 <- seq(0,2,1)
prob_2 <- c(1/2,1/4,1/4)
```

Si las dos variables son independientes, el mínimo valor que pueden tomar $x_1 + x_2$ es 0. Por otro lado, su máximo valor es 5. De esta forma, la suma puede tomar 6 valores diferentes: 0,1,2,3,4,5.

Pasamos ahora a calcular su tabla de probabilidad. Para ello, creamos dos tibble, uno con las probabilidades de cada uno de los valores de la suma, y otro con dichos valores:

```
(x <- merge(prob,prob_2,by = NULL) %>%
  mutate(probabilidad = x*y) %>%
  select(probabilidad))
```

```
##     probabilidad
## 1          0.256
```

```

## 2      0.192
## 3      0.048
## 4      0.004
## 5      0.128
## 6      0.096
## 7      0.024
## 8      0.002
## 9      0.128
## 10     0.096
## 11     0.024
## 12     0.002

(y <- merge(valor, valor_2, by = NULL) %>%
  mutate(valores = x + y) %>%
  select(valores))

```

```

##   valores
## 1      0
## 2      1
## 3      2
## 4      3
## 5      1
## 6      2
## 7      3
## 8      4
## 9      2
## 10     3
## 11     4
## 12     5

```

A partir de estos, la tabla de probabilidad es:

```

(val_prob <- as_tibble(c(x,y)) %>%
  mutate(probabilidad = probabilidad/sum(probabilidad)) %>%
  group_by(valores) %>%
  summarise(prob = sum(probabilidad)))

```

```

## # A tibble: 6 x 2
##   valores  prob
##   <dbl> <dbl>
## 1 0 0.256
## 2 1 0.32
## 3 2 0.272
## 4 3 0.124
## 5 4 0.026
## 6 5 0.002

```

Apartado 4: Calcula la media teórica de la suma $X_1 + X_2$. Después usa `sample` y `replicate` para simular cien mil *valores* de esta variable suma. Calcula la media de esos valores. *Advertencia:* no es el mismo tipo de análisis que hemos hecho en el segundo apartado.

Podemos calcular la media teórica utilizando la fórmula

$$\mu = \sum_{i=0}^n x_i p_i$$

```
(media_suma <- val_prob %>%
  summarise(media = sum(valores*prob)) %>%
  .[1,1])
```

```
## # A tibble: 1 x 1
##   media
##   <dbl>
## 1 1.35
```

O con la suma de ambas medias:

```
media_teorica_2 <- valor_2%*%prob_2 %>%
  .[1,1]

(media_suma_2 <- media_teorica + media_teorica_2)

## [1] 1.35
```

Con ello, vamos a obtener k muestras de la suma y vamos a calcular su media:

```
k = 100000

valores_suma <- replicate(k, {
  sample(val_prob$valores, 1, replace = TRUE, val_prob$prob)
})

mean(valores_suma)

## [1] 1.35328
```

El resultado al que llegamos es claro: la media de la suma es igual que la media de las muestras de la suma.

Ejercicio 2. Datos limpios

- Descargamos el Fichero de datos.
- Este fichero contiene las notas de los alumnos de una clase, que hicieron dos tests cada semana durante cinco semanas. La tabla de datos no cumple los principios de *tidy data* que hemos visto en clase. Tu tarea en este ejercicio es explicar por qué no se cumplen y obtener una tabla de datos limpios con la misma información usando *tidyR*.

Indicación: lee la ayuda de la función `separate` de *tidyR*.

```
notas <- read_csv(file = "./data/testResults.csv")

head(notas)

## # A tibble: 6 x 9
##   name      id gender_age test_number week1 week2 week3 week4 week5
##   <chr>    <dbl> <chr>        <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
```

```

## 1 Jacob      108 m_20          1     8     5     7     5     6
## 2 Jacob      108 m_20          2     2     2     4     0     3
## 3 Michael    490 m_19          1    10     0     5     4     0
## 4 Michael    490 m_19          2     9    10     8    10     9
## 5 Matthew    424 m_18          1     6     0     0     1    10
## 6 Matthew    424 m_18          2     3     4     2     5     8

```

Los principios del *tidy data* son claros :

1. Cada variable en una columna.
2. Cada observación en una fila.
3. Cada valor en una celda.

El fichero notas no es *tidy data*, pues, por un lado, las variables gender y age están en la misma columna, mientras que por otro lado, week debería ser una variable categórica cuyos posibles valores vayan desde week1 hasta week5. Es decir, tendremos que dividir la columna gender_age en gender y age (para tener dos columnas con su propia variable) y unificar las columnas de week en una sola con este nombre.

De esta forma, reordenamos los datos:

```

notas_2 <- notas %>%
  as_tibble() %>%
  pivot_longer(c("week1", "week2", "week3", "week4", "week5"), names_to = "week", values_to = "mark") %>%
  separate(gender_age, into = c("gender", "age"), sep = "_")

```

Comprobamos ahora que los datos sí son *tidy data*.

```
head(notas_2)
```

```

## # A tibble: 6 x 7
##   name    id gender age  test_number week  mark
##   <chr> <dbl> <chr> <chr>     <dbl> <chr> <dbl>
## 1 Jacob    108 m     20        1 week1     8
## 2 Jacob    108 m     20        1 week2     5
## 3 Jacob    108 m     20        1 week3     7
## 4 Jacob    108 m     20        1 week4     5
## 5 Jacob    108 m     20        1 week5     6
## 6 Jacob    108 m     20        2 week1     2

```

Ejercicio 3. Lectura de R4DS.

Continuando con nuestra *lectura conjunta* de este libro, si revisas el índice verás que hemos cubierto (holgadamente en algún caso) el contenido de los Capítulos 6, 8, 9, 10 y 11. Todos esos Capítulos son relativamente ligeros. Por eso esta semana conviene detenerse un poco en la lectura de los Capítulos 7 y 12, que son los más densos en información. Y como motivación os proponemos un par de ejercicios, uno por cada uno de esos capítulos.

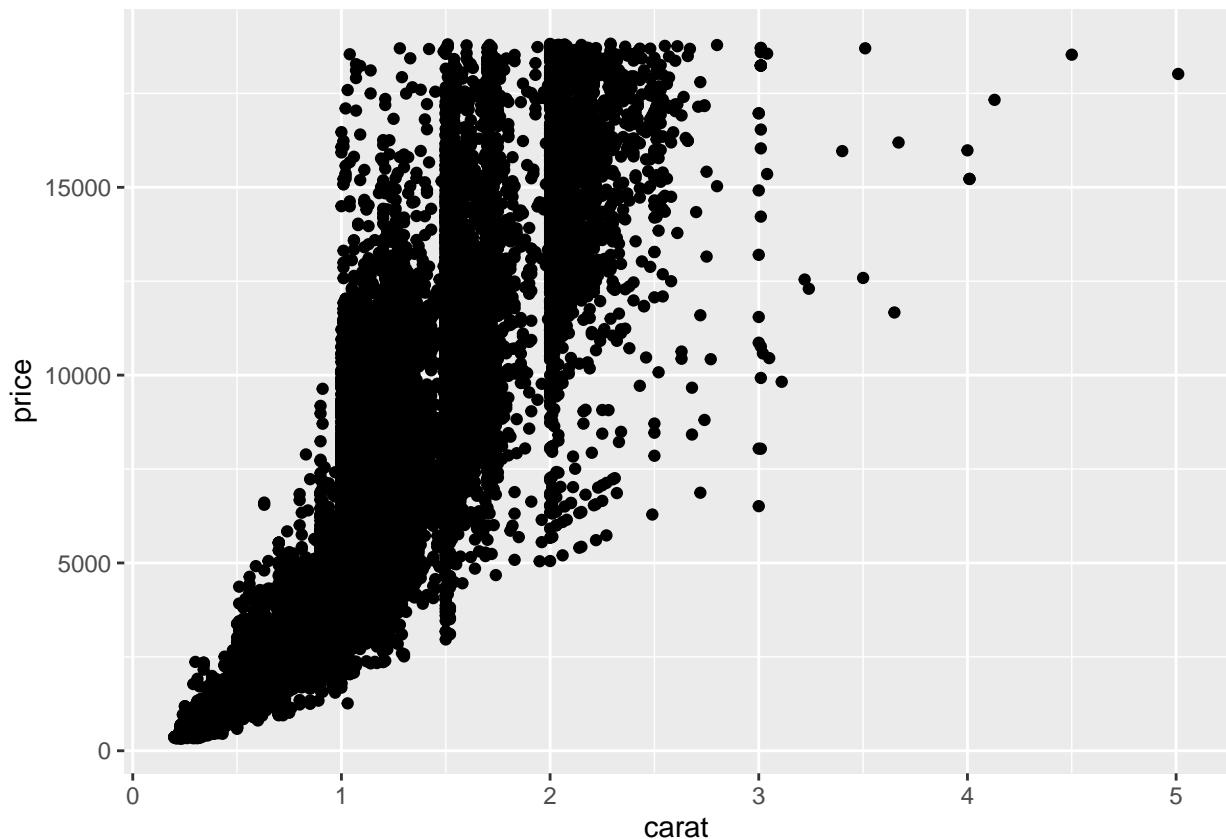
Ejercicio 2 de la sección 7.5.1.1:

¿Qué variable es más importante en el dataset diamonds para predecir el precio de un diamante? ¿Cómo está esa variable relacionada con cut? ¿Por qué la combinación de esas dos relaciones lleva a que los diamantes de menor calidad son más caros?

Utilizamos el dataset de R diamonds. De todas las variables de la tabla, tenemos 3 que son factores: cut, color y clarity. El resto son variables numéricas continuas. Para ver cuál influye más en el precio de un diamante, vamos a representar, frente al precio, las diferentes variables.

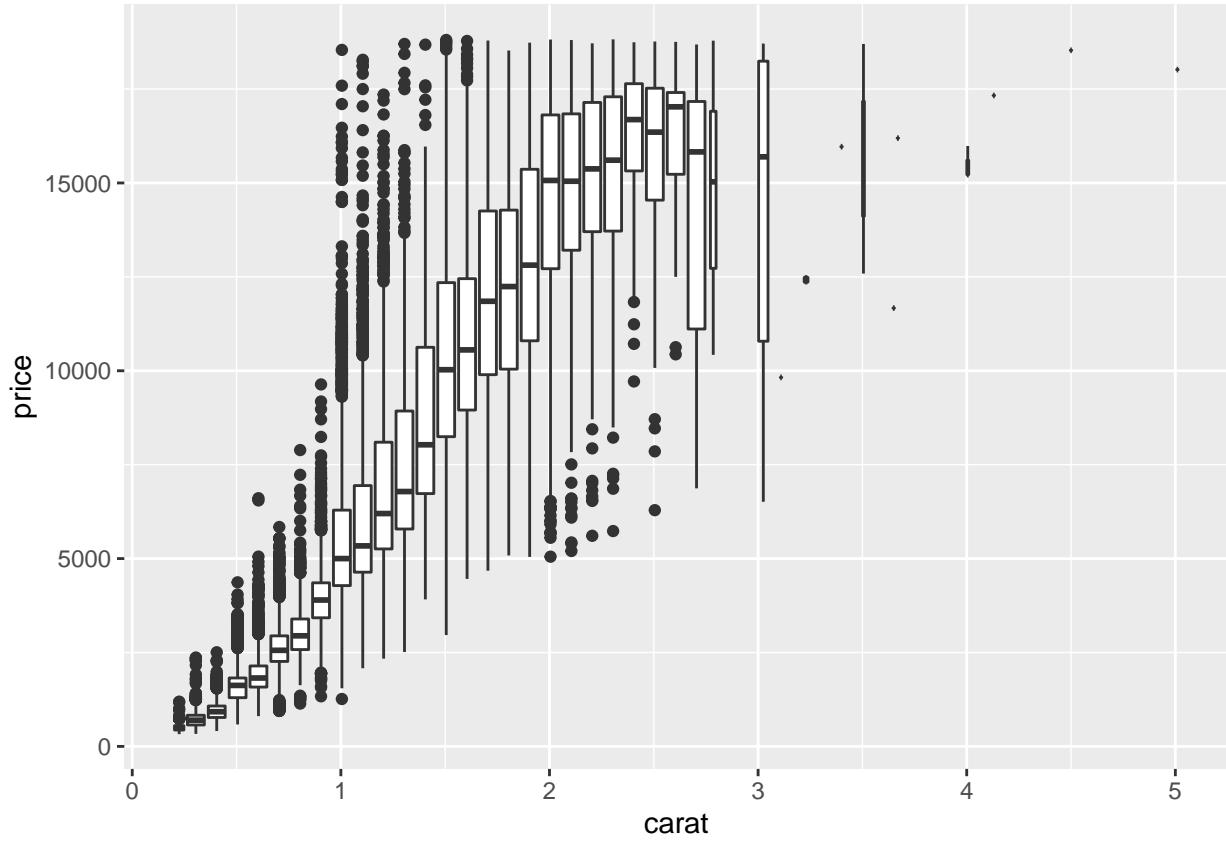
En primer lugar, representamos en un diagrama de puntos del precio frente a carat.

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



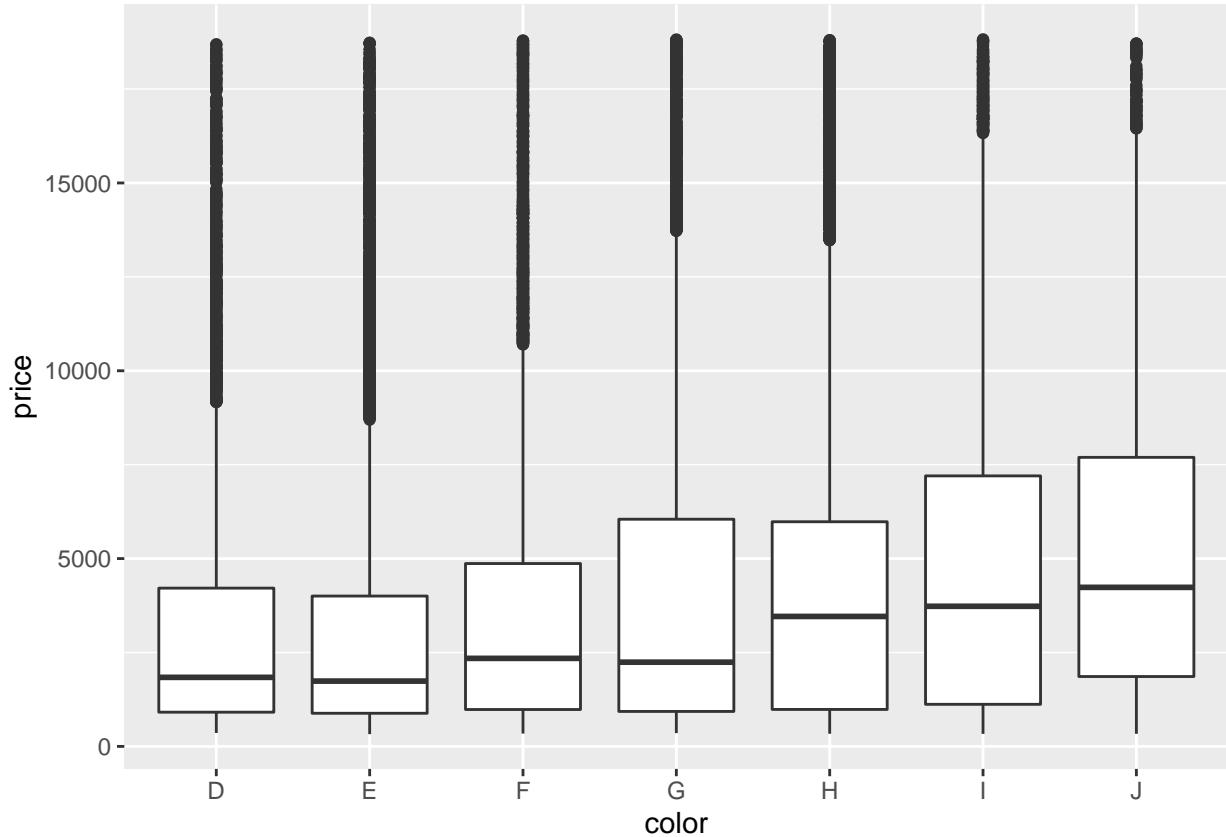
Vemos que existe una fuerte correlación: a medida que aumenta carat, aumenta en gran medida el precio de los diamantes. Pero vamos a representar un boxplot con esta misma información con tal de que se represente de forma más clara, agrupada por el carat de los diamantes.

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



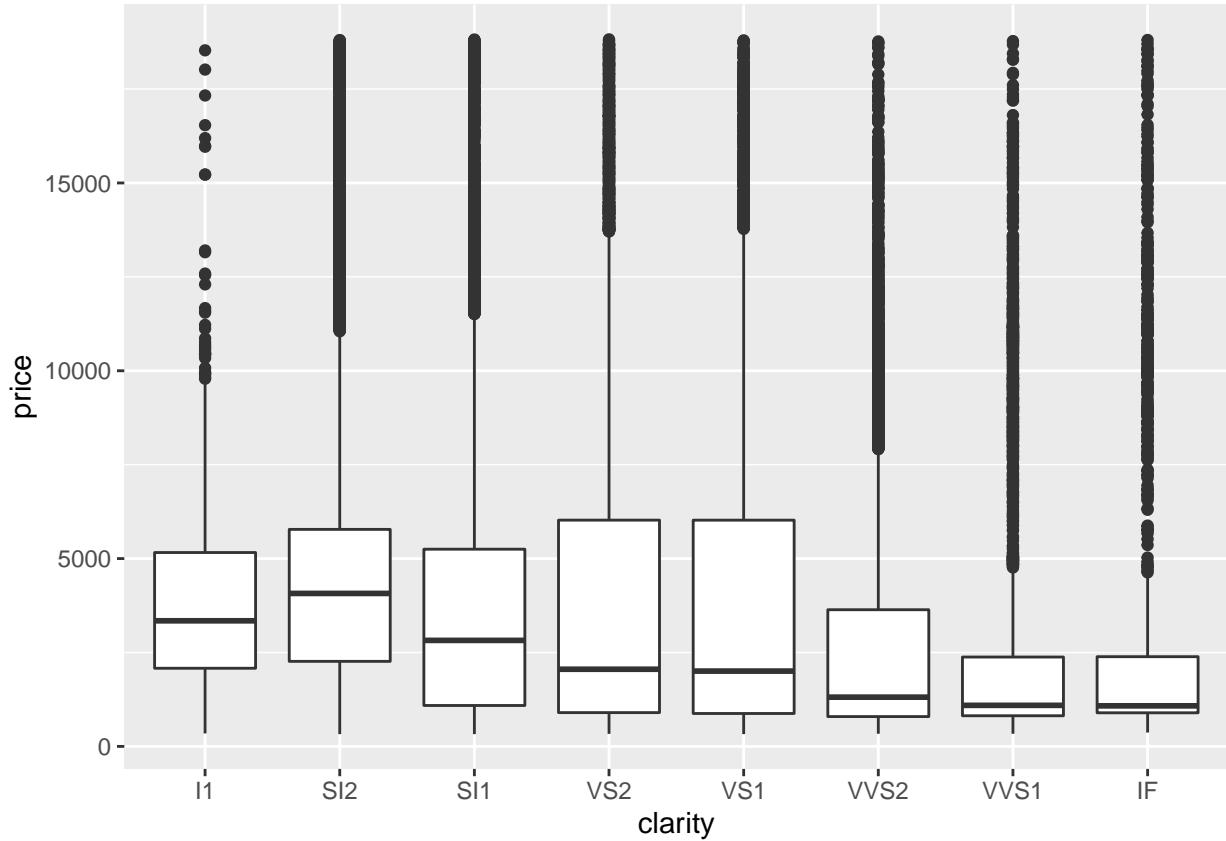
Volvemos a observar claramente que, al aumentar el carat de los diamantes, estos aumentan su precio. Sin embargo, hemos de contemplar el resto de variables antes de afirmar que es carat la más importante a la hora de determinar el precio de los diamantes. Utilizamos un boxplot para representar variables continuas (en este caso el precio) frente a variables categóricas. En primer lugar, vamos a ver cómo se comporta el precio frente al color de los diamantes.

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



Según la descripción de RStudio de la tabla diamonds, el mejor color es el D y el peor el J. Sin embargo, vemos una relación inversa en el precio con respecto a la calidad de los colores, pues los más caros son los J y los más baratos los de color D. A pesar de ello, la variación del precio respecto al color es mucho menor que con respecto a la variable carat, tal y como se comprueba comparando ambas gráficas. Ahora veremos la relación entre el precio y la claridad de los diamantes.

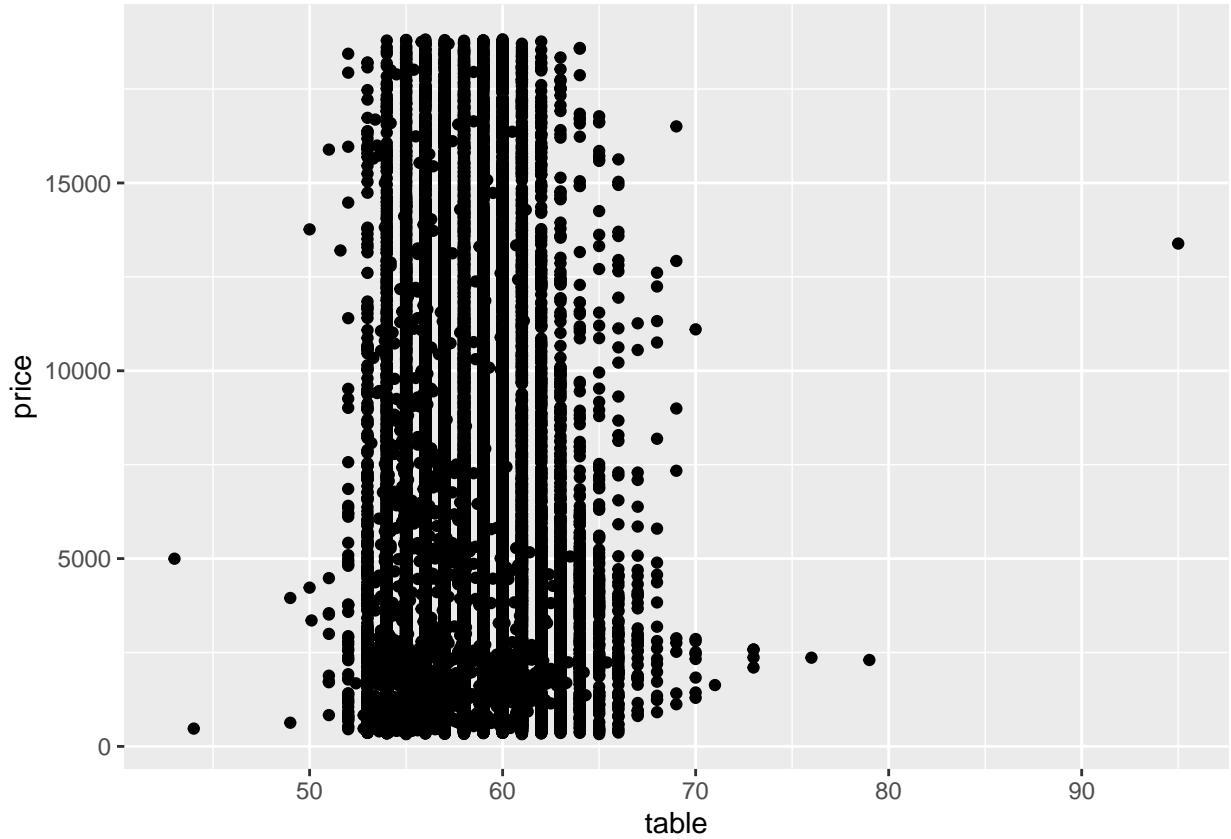
```
ggplot(data = diamonds) +
  geom_boxplot(mapping = aes(x = clarity, y = price))
```



Volvemos a ver que la variación del precio con clarity no es significativa. De hecho, tanto esta variable como color, varían mucho más el precio dentro de sus propias categorías que entre categorías (esto es, el IQR es mayor que la variación del precio entre diferentes valores de clarity).

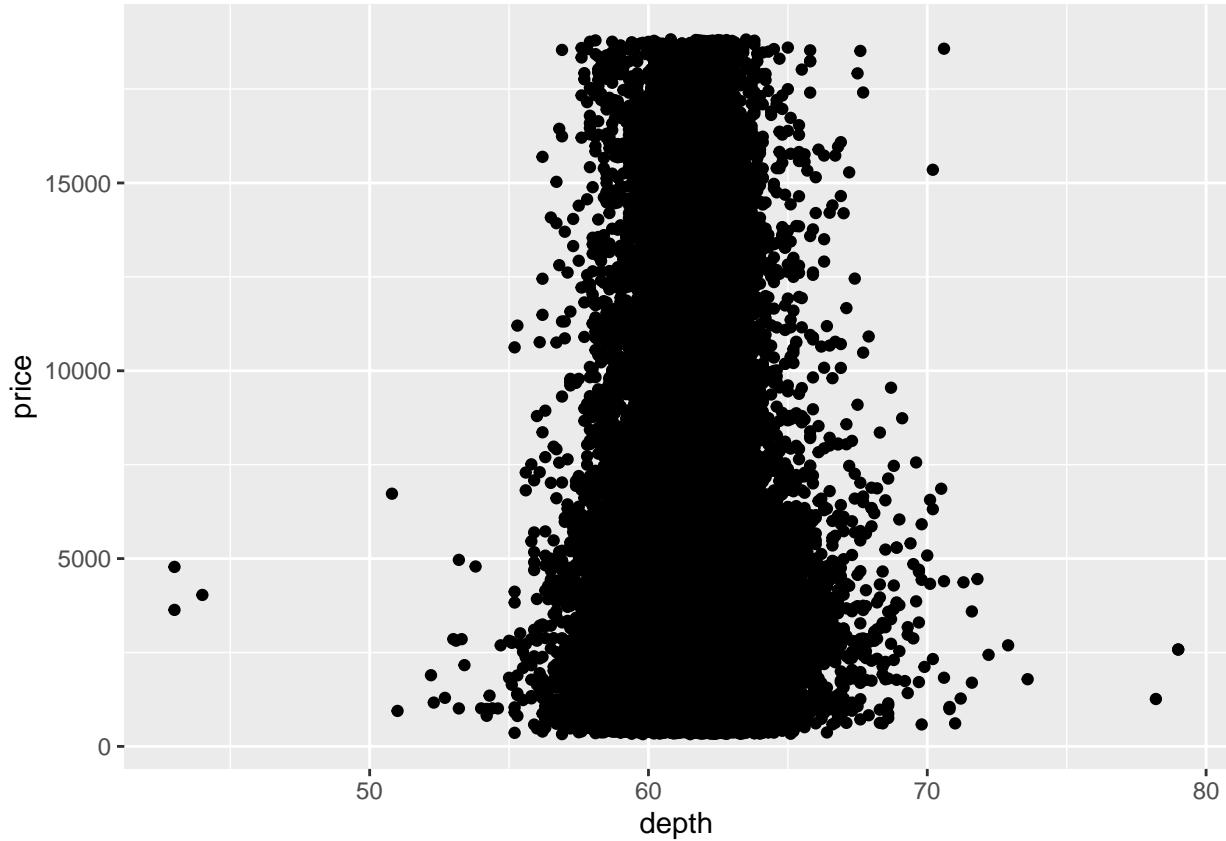
Las variables que faltan por comprobar son table y depth (pues considerando esas dos consideramos todas las dimensiones):

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



Vemos que no se observa ninguna relación porque hay diamantes de todos los precios independientemente de la variable table.

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

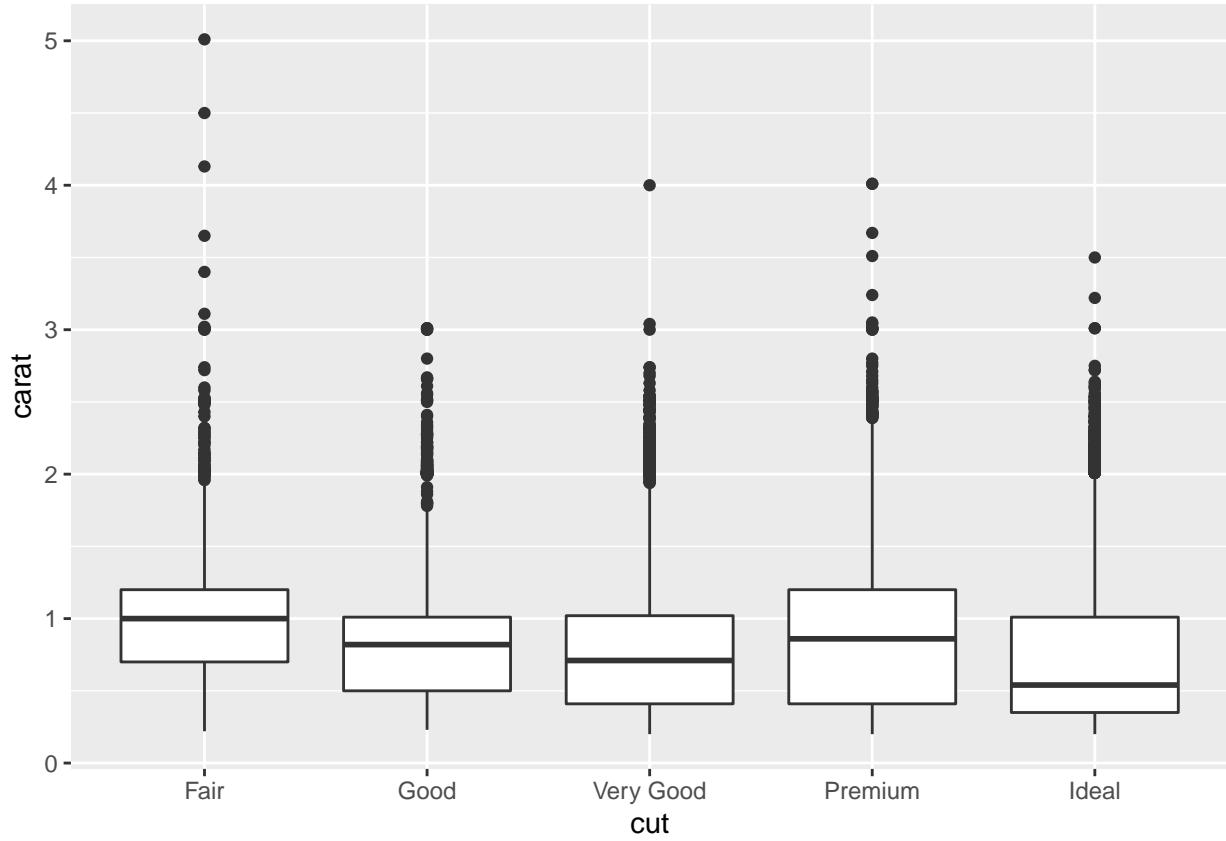


Tampoco observamos ninguna relación evidente en el caso de la profundidad de los diamantes.

Con todo ello, podemos llegar a la conclusión de que, efectivamente, la variable más importante a la hora de determinar el precio de los diamantes es carat.

Ahora vamos a ver cuál es la relación entre cut y carat.

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



Se observa que conforme aumenta la calidad del corte de los diamantes (derecha en el eje x), aumenta la variabilidad en los quilates de estos (tienen un IQR mayor). Además, hay una ligerísima relación negativa entre el tamaño y el corte: los diamantes tipo Fair tienen más quilates y los Ideal, menos. Por último, vemos que hay muchos datos atípicos por encima pero ninguno por debajo.

Tal y como dice el enunciado, es cierto que los diamantes cuyo corte tiene menor calidad tienen mayor carat, y por lo tanto mayor precio (ya hemos discutido que el precio aumenta con el carat). Este resultado es contradictorio. Sin embargo, se puede entender si nos fijamos en los valores atípicos. Los diamantes de corte Fair tienen más valores atípicos por encima que el resto de cortes. Esto implica que su media esté desplazada hacia arriba respecto al resto de cortes: tiene más desviación.

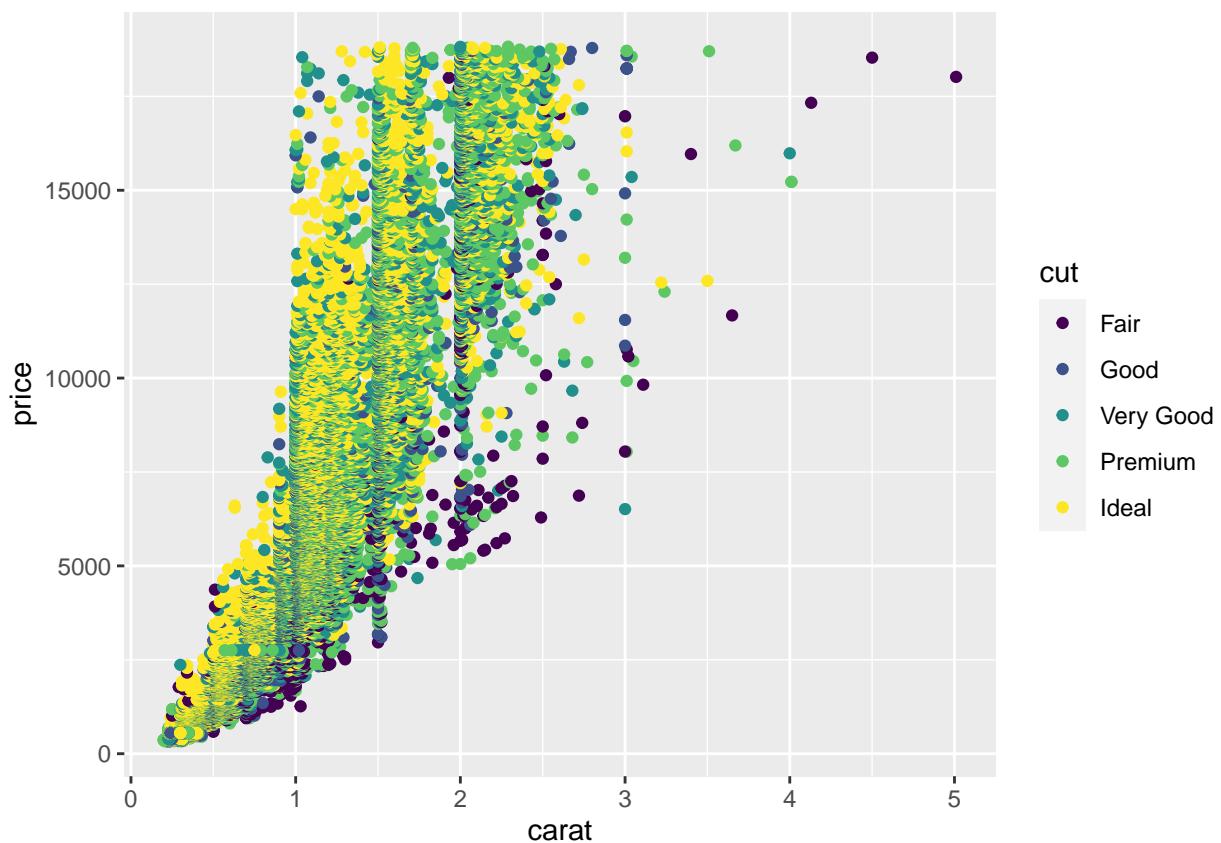
Podemos comprobarlo:

```
diamonds %>%
  group_by(cut) %>%
  summarise(sd(carat))

## # A tibble: 5 x 2
##   cut      `sd(carat)`
##   <ord>        <dbl>
## 1 Fair       0.516
## 2 Good      0.454
## 3 Very Good 0.459
## 4 Premium    0.515
## 5 Ideal      0.433
```

También podemos ver, si volvemos a representar el precio frente a carat, como los diamantes de tipo Fair son los más baratos:

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



Es decir, que la mediana de los quilates del corte Fair sea mayor que el resto, al ser tan pequeña la diferencia, no nos sirve para determinar que su precio es mayor, si no que se lo atribuimos a sus valores atípicos.

Ejercicio 4 de la Sección 12.6.1:

Para cada país, año y sexo computar el número de casos totales de TB. Haz una visualización informativa de los datos.

Tal y como se nos indica en la práctica, vamos a utilizar el código limpio que aparece en la sección 12.6.1 de R4DS.

```
who1 <- who %>%
  pivot_longer(
    cols = new_sp_m014:newrel_f65,
    names_to = "key",
    values_to = "cases",
    values_drop_na = TRUE
  )

who2 <- who1 %>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel"))

who3 <- who2 %>%
```

```

separate(key, c("new", "type", "sexage"), sep = "_")

who4 <- who3 %>%
  select(-new, -iso2, -iso3)

who5 <- who4 %>%
  separate(sexage, c("sex", "age"), sep = 1)

```

Ahora realizamos el ejercicio.

```

who5 %>%
  group_by(country, year, sex) %>%
  summarise(cases = sum(cases))

```

```

## # A tibble: 6,921 x 4
## # Groups:   country, year [3,484]
##       country     year   sex   cases
##       <chr>      <int> <chr> <int>
## 1 Afghanistan  1997   f     102
## 2 Afghanistan  1997   m      26
## 3 Afghanistan  1998   f    1207
## 4 Afghanistan  1998   m      571
## 5 Afghanistan  1999   f      517
## 6 Afghanistan  1999   m      228
## 7 Afghanistan  2000   f    1751
## 8 Afghanistan  2000   m      915
## 9 Afghanistan  2001   f    3062
## 10 Afghanistan 2001   m     1577
## # ... with 6,911 more rows

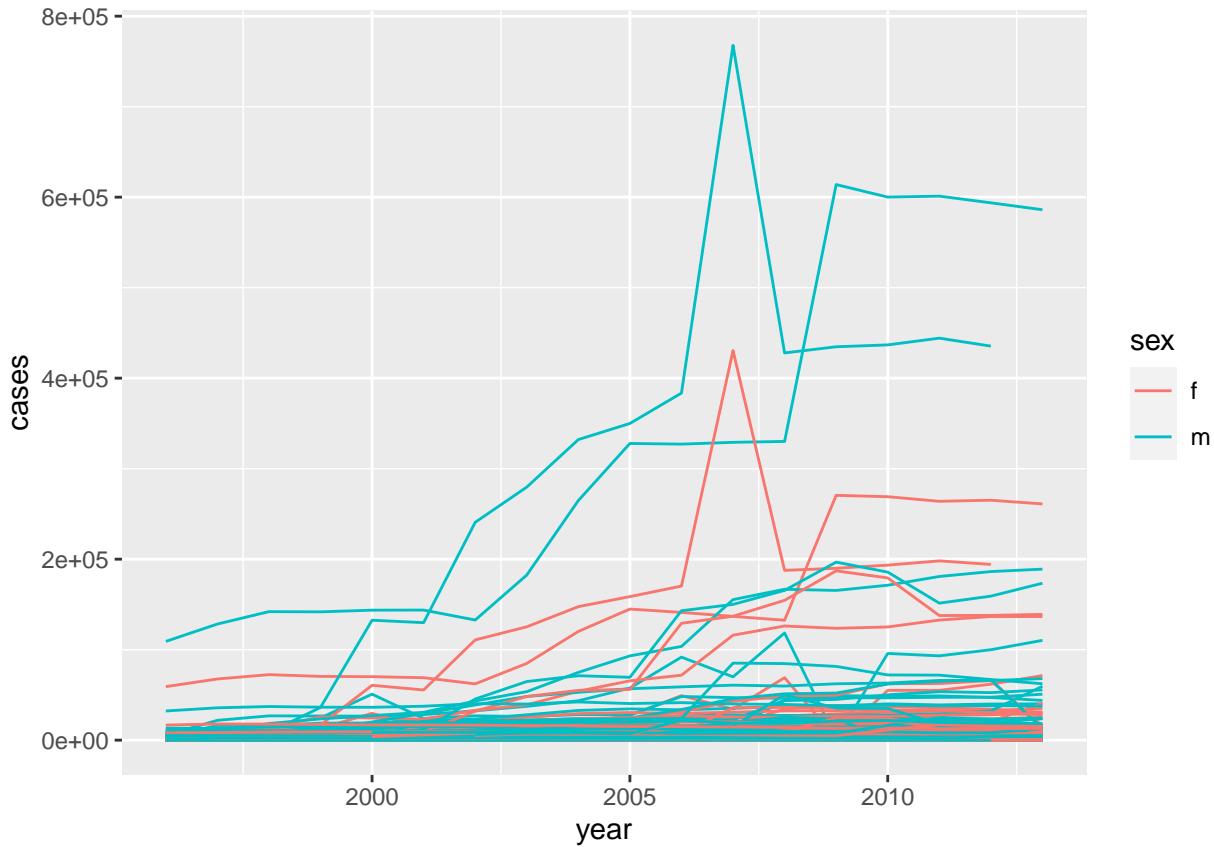
```

A partir de ahora, para realizar las representaciones gráficas, consideraremos los años a partir de 1995 porque en los años anteriores no hay mucha información.

```

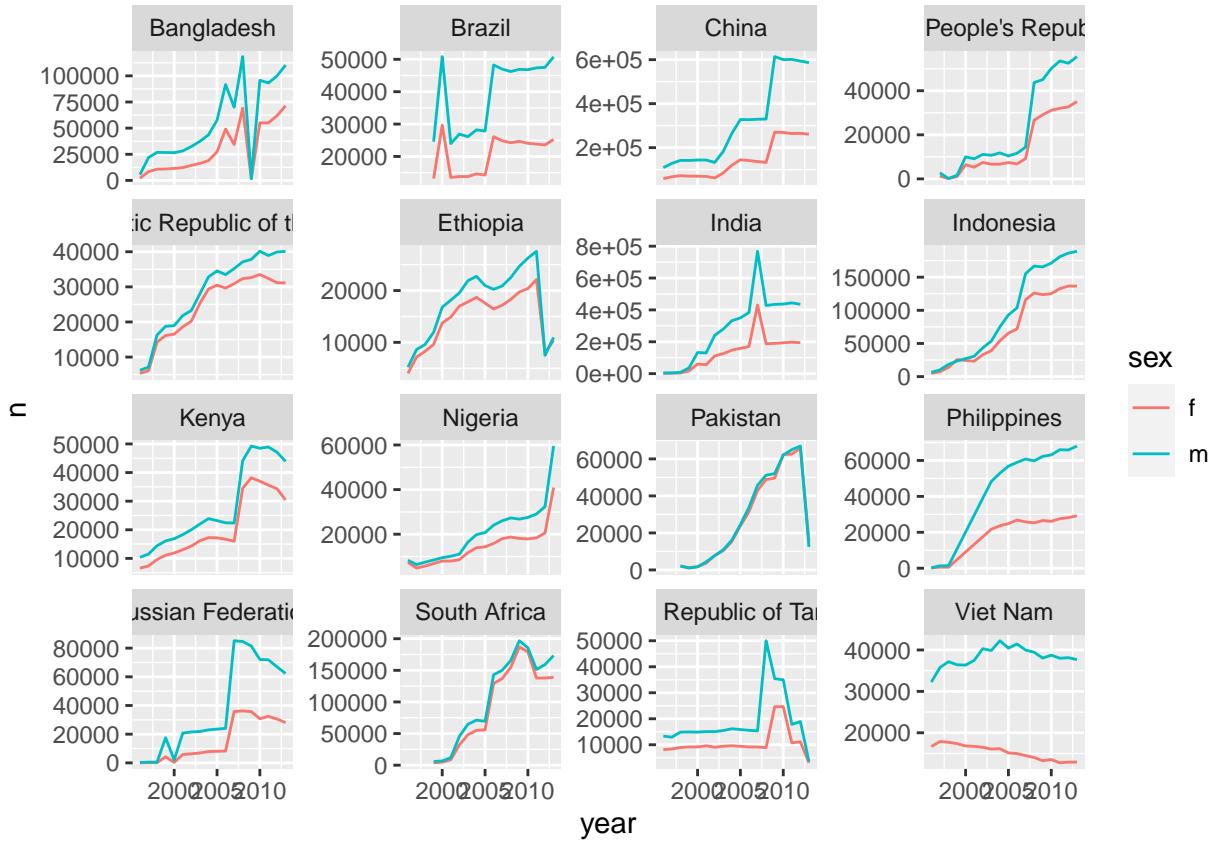
who5 %>%
  group_by(country, year, sex) %>%
  filter(year > 1995) %>%
  summarise(cases = sum(cases)) %>%
  unite(country_sex, country, sex, remove = FALSE) %>%
  ggplot(aes(x = year, y = cases, group = country_sex, colour = sex)) +
  geom_line()

```



Esta gráfica no es muy clara, y vemos que en la parte de abajo hay muchos países, por lo que podemos analizar diferentes cosas, como los países con más casos de tuberculosis y separarlos por hombre y mujer:

```
who5 %>%
  group_by(country, sex, year) %>%
  filter(year>1995) %>%
  summarise(n=sum(cases)) %>%
  ungroup() %>%
  group_by(country) %>%
  mutate(total_country=sum(n)) %>%
  filter(total_country>500000) %>%
  ggplot(aes(x=year,y=n,colour=sex))+
  geom_line()+
  facet_wrap(~country, scales = "free_y" )
```



En este conjunto de gráficos se representa la información de los países con más de 500000 casos de tuberculosis, separados en hombre y mujer. Podemos observar que en la mayoría de ellos, el número de casos en hombres es mayor que en mujeres.