

# Tarea 2

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

Pablo Soriano González

21/09/2021

## Pasos preliminares:

Comenzamos cargando librerías:

```
library(tidyverse)
library(dplyr)
library(ggplot2)
```

## 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.

Creamos dos vectores, uno con los posibles valores que puede tomar nuestra variable aleatoria y otro con las probabilidades de cada uno de ellos.

```
# Valores
(valx1 <- seq(0,3,1))

## [1] 0 1 2 3

# Probabilidades
(probxb1 <- c(64,48,12,1)/125)

## [1] 0.512 0.384 0.096 0.008
```

Para calcular la media teórica de esta variable tenemos que multiplicar cada uno de los posibles valores por la probabilidad de que aparezca y sumar todo:

$$\mu = x_1 p_1 + \cdots + x_k p_k$$

Para ello empleamos el producto escalar de dos vectores entre los vectores anteriores. Como el resultado es una tabla [1x1] debemos acceder a su contenido para obtener un escalar.

```
(mediax1 <- (valx1%*%probxb1 %>%
  .[1,1]))

## [1] 0.6
```

Para la varianza teórica de la variable debemos restar a cada uno de los posibles valores de la variable la media, elevar esa diferencia al cuadrado, multiplicarlo por su respectiva probabilidad y sumar todo:

$$\sigma^2 = (x_1 - \mu)^2 p_1 + \cdots + (x_k - \mu)^2 p_k$$

Lo hacemos de forma vectorial y hacemos un sum del vector:

```
(desvx1 <- sum(((valx1-mediamx1)^2)*probx1))
```

```
## [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:

Definiremos como nos han dicho el número de veces del replicate (k) como 100000 y crearemos las muestras y sus respectivas medias muestrales. Comenzamos con muestras de tamaño 10:

```
# Definimos la cantidad de muestras que crearemos cada vez
k=100000
```

```
# Creamos las primeras medias muestrales a partir de las muestras aleatorias de tamaño 10
muestrasx1 <- replicate(k,mean(sample(valx1,10,replace = TRUE,probx1)))
```

```
# Observamos varias de estas medias muestrales
head(muestrasx1, 10)
```

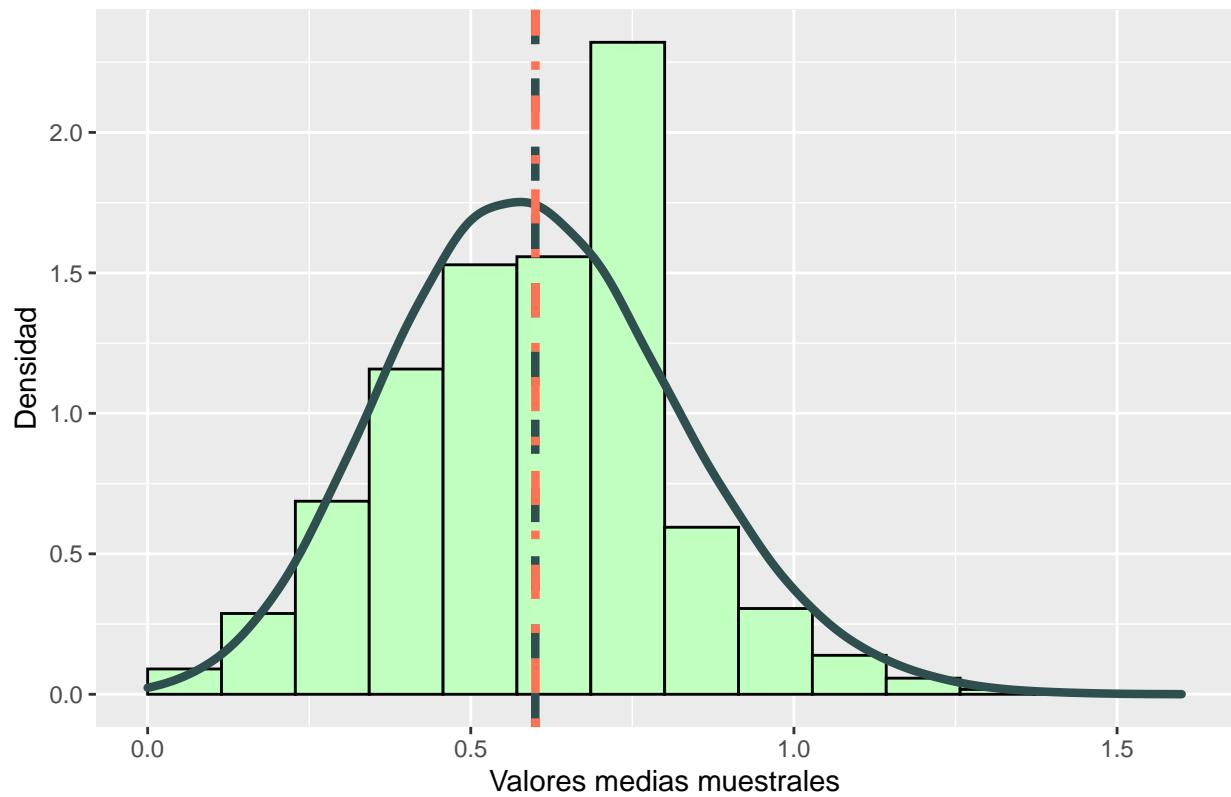
```
## [1] 0.4 0.7 0.7 0.5 0.9 0.4 0.3 0.2 0.6 0.7
```

Realizamos la representación gráfica de estas medias muestrales mediante un histograma, su curva de densidad, la media teórica de nuestra variable (color coral) y la media de nuestras medias muestrales (gris oscuro):

```
cortes = seq(min(muestrasx1), max(muestrasx1), length.out = 15)

ggplot(as.data.frame(muestrasx1), aes(x = muestrasx1)) +
  geom_histogram(aes(y=stat(density)),
    breaks = cortes, fill = "darkseagreen1", color="black") +
  geom_density(color="darkslategrey", size=1.5, adjust = 3) +
  geom_vline(xintercept = mean(muestrasx1), lty=2, lwd=1.5, col="darkslategrey") +
  geom_vline(xintercept = mediamx1, lty=4, lwd=1.5, col="coral1") +
  ggtitle("Distribución de medias muestrales con muestras n = 10") +
  ylab("Densidad")+
  xlab("Valores medias muestrales")
```

## Distribución de medias muestrales con muestras n = 10



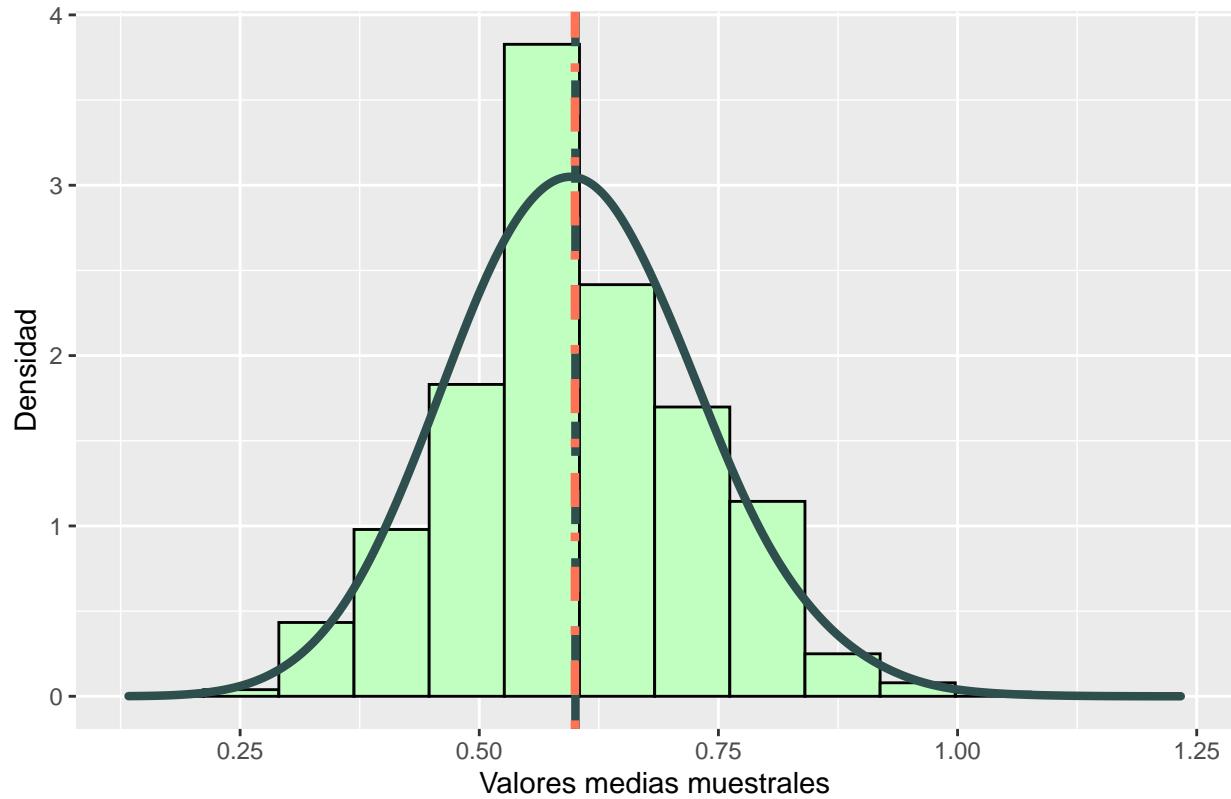
Repetimos ahora el proceso pero tomando muestras de tamaño 30:

```
muestras2x1 <- replicate(k,mean(sample(valx1,30,replace = TRUE,prob=x1)))

cortes = seq(min(muestras2x1), max(muestras2x1), length.out = 15)

ggplot(as.data.frame(muestras2x1), aes(x = muestras2x1)) +
  geom_histogram(aes(y=stat(density)),
                 breaks = cortes, fill = "darkseagreen1", color="black") +
  geom_density(color="darkslategrey", size=1.5, adjust = 3) +
  geom_vline(xintercept = mean(muestras2x1), lty=2, lwd=1.5, col="darkslategrey") +
  geom_vline(xintercept = mediax1, lty=4, lwd=1.5, col="coral1") +
  ggtitle("Distribución de medias muestrales con muestras n = 30") +
  ylab("Densidad")+
  xlab("Valores medias muestrales")
```

## Distribución de medias muestrales con muestras n = 30



Vemos como tanto el histograma como la curva de densidad se asemejan más a una curva normal ahora que antes, el máximo del histograma coincide mejor con la media teórica de la variable y con la media de las medias muestrales.

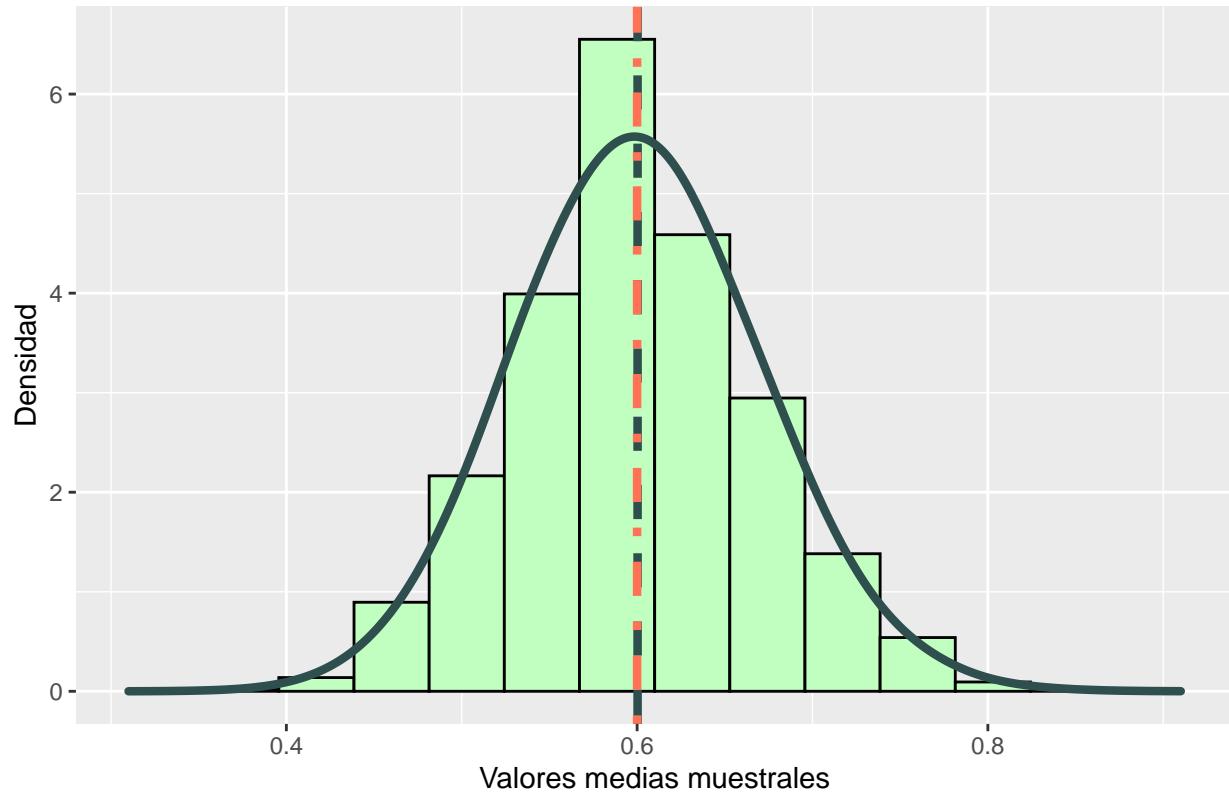
Si finalmente aumentamos el tamaño de cada una de las muestras a 100 obtenemos lo siguiente:

```
muestras3x1 <- replicate(k, mean(sample(valx1, 100, replace = TRUE, probx1)))

cortes = seq(min(muestras3x1), max(muestras3x1), length.out = 15)

ggplot(as.data.frame(muestras3x1), aes(x = muestras3x1)) +
  geom_histogram(aes(y=stat(density)),
                 breaks = cortes, fill = "darkseagreen1", color="black") +
  geom_density(color="darkslategrey", size=1.5, adjust = 3) +
  geom_vline(xintercept = mean(muestras3x1), lty=2, lwd=1.5, col="darkslategrey") +
  geom_vline(xintercept = mediax1, lty=4, lwd=1.5, col="coral1") +
  ggtitle("Distribución de medias muestrales con muestras n = 100") +
  ylab("Densidad")+
  xlab("Valores medias muestrales")
```

## Distribución de medias muestrales con muestras n = 100



Vemos como con este tamaño de muestras tanto el histograma como la curva de densidad se ajustan muy bien a una curva normal, tal y como cabría esperar.

**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 $P(X = x_i)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$

Creamos dos vectores que almacenen tanto los posibles valores de la variable como las probabilidades de cada uno de ellos:

```
valx2 <- seq(0,2,1)
probx2 <- c(0.5,0.25, 0.25)
```

**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?**

Vemos de forma sencilla que los valores posibles de la suma van desde obtener 0 en ambas variables y, por tanto, un 0 en la suma a obtener los valores máximos, 3 y 2, y un 5 en la suma, es decir, (0,1,2,3,4,5). No obstante, deberíamos poder automatizar esto para poder calcularlo en casos más complicados.

Para ello creamos primero una matriz vacía que tendrá en las columnas los posibles valores de una de las variables y en la otra los posibles valores de la otra variable. Despues rellenamos esta matriz de manera que los elementos sean la suma de los correspondientes elementos de cada variable:

```
# Creación de la matriz vacía
posValSum <- matrix(nrow = 4, ncol = 3, dimnames = list(c("0","1","2","3"), c("0","1","2")))
```

```

# La rellenamos
for(i in valx1){
  for(j in valx2){
    posValSum[i+1,j+1] <- i+j
  }
}

# Y la visualizamos
posValSum

```

```

## 0 1 2
## 0 0 1 2
## 1 1 2 3
## 2 2 3 4
## 3 3 4 5

```

Finalmente, para saber cuáles son los posibles valores de la suma usamos el comando `unique()`, que nos proporciona un vector que contiene cada uno de los valores del vector una única vez. Para ello pasamos a vector esta matriz con `as.vector()`:

```
(Suma <- unique(as.vector(posValSum)))
```

```
## [1] 0 1 2 3 4 5
```

Para obtener la tabla de probabilidad de estos resultados de la suma tenemos que tener en cuenta que la suma es commutativa, por lo que la probabilidad de obtener un resultado en la operación será la suma de las probabilidades de obtenerlo de las distintas formas que existan. Por ejemplo, el 2 lo podremos obtener de 3 formas distintas  $2+0$ ,  $0+2$  y  $1+1$ . Teniendo esto en cuenta creamos un vector vacío para almacenar las probabilidades de cada uno de los distintos valores posibles de la suma y lo rellenamos empleando un bucle que recorra las diferentes combinaciones:

```

probSuma <- rep(0,6)

for(i in Suma){
  for(j in valx1){
    for(k in valx2){
      if((j+k)==i){
        probSuma[i+1] <- probSuma[i+1] + probx1[j+1]*probx2[k+1]
      }
    }
  }
}

```

Observamos el vector de probabilidades y lo sumamos para comprobar que la probabilidad total suma 1:

```
# Vector de probabilidades
probSuma
```

```
## [1] 0.256 0.320 0.272 0.124 0.026 0.002
```

```
# Suma de las probabilidades
sum(probSuma)
```

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

Ahora generamos la tabla de probabilidad:

```
(tablaprobfinal <- matrix(probSuma, nrow=1, ncol = 6, byrow=TRUE,
                           dimnames = list(c("Probabilidades"), Suma)))
```

```

##          0      1      2      3      4      5
## Probabilidades 0.256 0.32 0.272 0.124 0.026 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.*

Empleando los vectores Suma y probSuma calculados en el apartado anterior calculamos la media teórica de los valores de la variable suma de la misma forma que hicimos en el primer apartado con la variable X1.

```
(mediaSuma <- (Suma %*% probSuma %>%
  .[1,1]))
```

```
## [1] 1.35
```

Ahora usamos los comandos `replicate` y `sample` para generar cien mil valores aleatorios de esta variable y calculamos su media:

```
# Generamos los valores
simulSuma <- replicate(100000, sample(Suma, 1, replace = TRUE, prob = probSuma))
```

```
# Visualizamos los primeros 20
head(simulSuma, 20)
```

```
## [1] 1 1 2 1 2 2 1 2 1 1 3 0 2 0 1 2 1 2 1 1
```

```
# Y calculamos la media
mean(simulSuma)
```

```
## [1] 1.35158
```

## Ejercicio 2. Datos limpios.

- Descarga el fichero de este enlace:

<https://gist.githubusercontent.com/fernandosansegundo/471b4887737cfcec7e9cf28631f2e21e/raw/b3944599d02df494f5903740db5acac9da35bc6f/testResults.csv>

- 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*.

Una vez realizada la descarga del fichero lo cargamos:

```
test <- read.csv(file = "data/testResults.csv", header = TRUE, sep = ",")
```

```

##      name  id gender_age test_number week1 week2 week3 week4 week5
## 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

```

Al ver la tabla de datos observamos distintas cosas que no cumplen los principios de tidy data. Por un lado, vemos que hay una columna que almacena tanto la información del género como la de la edad (`gender_age`), por lo que deberíamos separarla en dos columnas distintas que almacenen esta información por separado.

Esto lo haremos empleando el comando separate. También vemos que tenemos 5 columnas con datos, una para cada una de las semanas. Para cumplir con el tidy data deberíamos transformar estas columnas en dos únicamente: una que almacene la información de a qué semana hace referencia la medida y otra con el valor de la medida en si, teniendo así una única medida por cada fila, con su correspondiente información en el resto de columna. Para hacer esto usaremos pivot\_longer. Si solo hicieramos esto la columna de week sería una columna que almacenaría strings del estilo “week1”. Para evitar esto definimos la función *substrRight(x, n)* que nos da los n-últimos caracteres de un string y mutamos la columna sacando el último carácter de cada elemento de esta columna, es decir, el número de la semana y convirtiéndolo a número con as.numeric.

```
# Definimos la función que extrae los n-últimos caracteres de un string
substrRight <- function(x, n){
  substr(x, nchar(x)-n+1, nchar(x))
}

# Modificamos la tabla test y la almacenamos en testClean
testClean <- test %>%
  separate(gender_age, into = c("gender", "age"), sep = "_", convert = TRUE) %>%
  pivot_longer(c("week1", "week2", "week3", "week4", "week5"),
               names_to = "week", values_to = "medida") %>%
  mutate(week, week = as.numeric(substrRight(week, 1)))

# Mostramos los primeros valores de la tabla para ver el resultado
head(testClean)

## # A tibble: 6 x 7
##   name     id gender   age test_number week medida
##   <chr> <int> <chr> <int>      <int> <dbl> <int>
## 1 Jacob    108 m        20          1     1       8
## 2 Jacob    108 m        20          1     2       5
## 3 Jacob    108 m        20          1     3       7
## 4 Jacob    108 m        20          1     4       5
## 5 Jacob    108 m        20          1     5       6
## 6 Jacob    108 m        20          2     1       2
```

## Ejercicio 3. Lectura de R4DS.

Haz el ejercicio 2 de la Sección 7.5.1.1 de R4DS.

En este ejercicio se nos pide responder a las siguientes preguntas:

¿Qué variable en el dataset de los diamantes es más importante para predecir el precio de un diamante?  
 ¿Cómo se relaciona esa variable con “cut”? ¿Por qué la combinación de esas dos relaciones hace que los diamantes de más baja calidad sean más caros?

Si mostramos el dataset de los diamantes vemos su estructura:

```
head(diamonds)
```

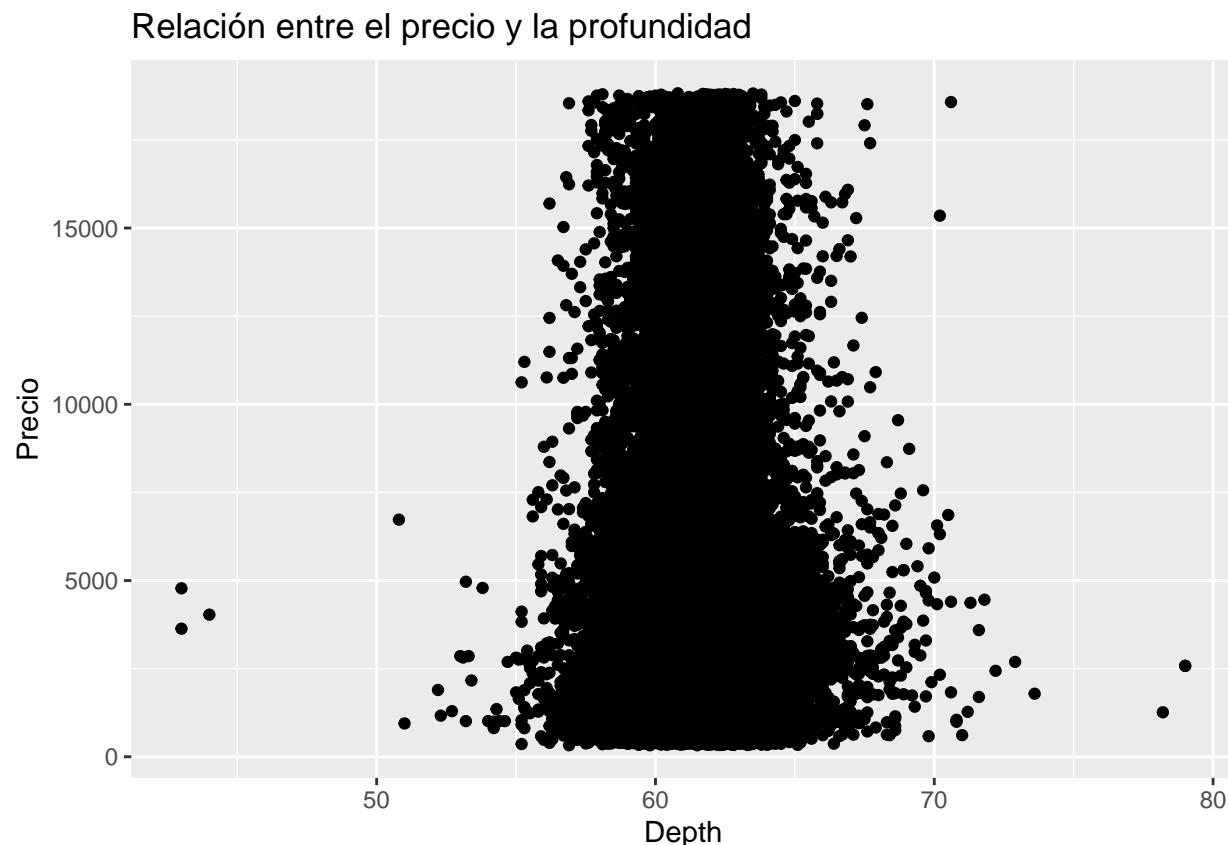
```
## # A tibble: 6 x 10
##   carat cut      color clarity depth table price     x     y     z
##   <dbl> <ord>    <ord> <ord>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  0.23 Ideal    E     SI2     61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium  E     SI1     59.8    61   326  3.89  3.84  2.31
## 3  0.23 Good     E     VS1     56.9    65   327  4.05  4.07  2.31
## 4  0.29 Premium  I     VS2     62.4    58   334  4.2   4.23  2.63
## 5  0.31 Good     J     SI2     63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2    62.8    57   336  3.94  3.96  2.48
```

Una lectura simple de la descripción de cada uno de los campos nos permite comprobar que hay variables que podemos dejar fuera de nuestro análisis. Por ejemplo, las dimensiones x, y, z, están directamente relacionadas con el peso del diamante, es decir, los quilates, analizados en “carat” o también con la profundidad, “depth”. Tenemos, por tanto, que analizar las relaciones entre “carat”, “cut”, “color”, “clarity”, “depth” y “table” con el precio “price”.

Para el análisis de las variables continuas (carat, depth y table) emplearemos diagramas de dispersión mientras que para las categóricas (cut, color y clarity) usaremos boxplots.

## DEPTH

```
ggplot(diamonds, aes(x = depth, y = price)) +
  geom_point() +
  ggtitle("Relación entre el precio y la profundidad") +
  ylab("Precio") +
  xlab("Depth")
```

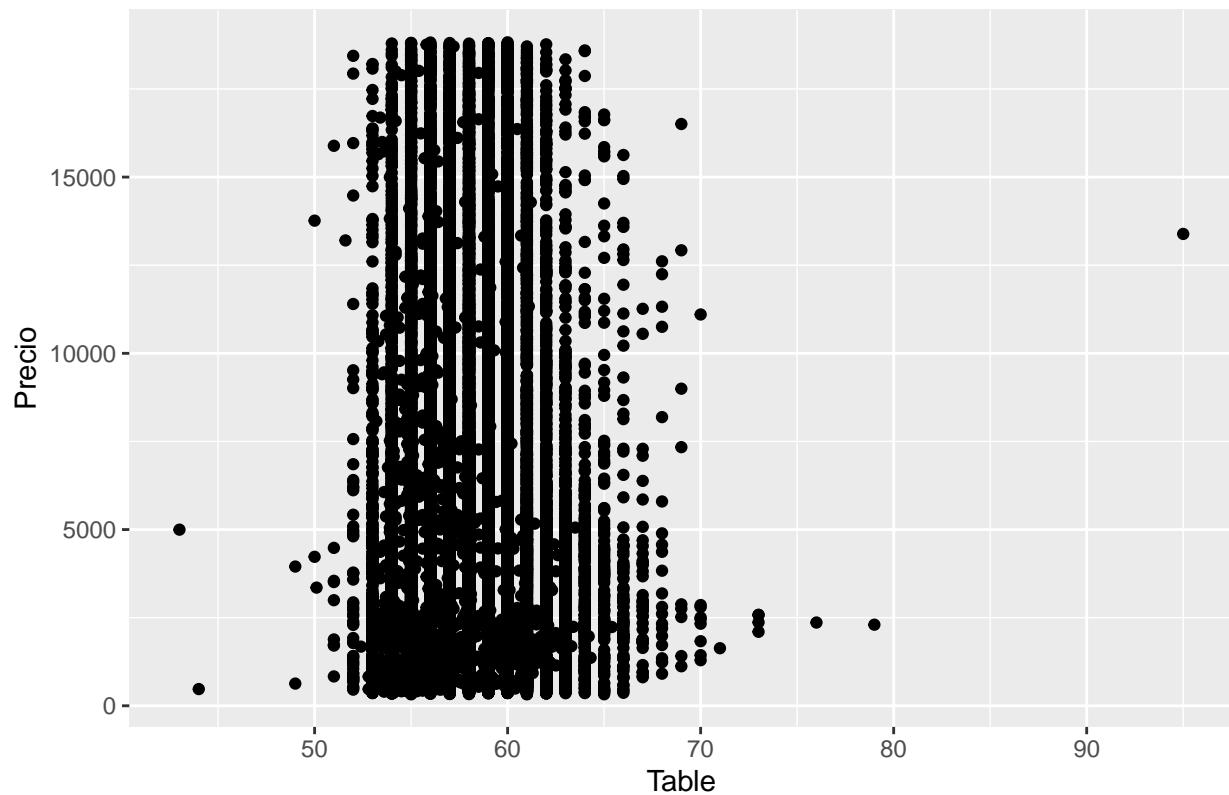


En el diagrama observamos como las profundidades toman valores principalmente entre 55 y 65 y que no existe una correlación con el precio pues existen diamantes de todo el rango de precios de distintas profundidades en este rango.

## TABLE

```
ggplot(diamonds, aes(x = table, y = price)) +
  geom_point() +
  ggtitle("Relación entre el precio y table") +
  ylab("Precio") +
  xlab("Table")
```

## Relación entre el precio y table

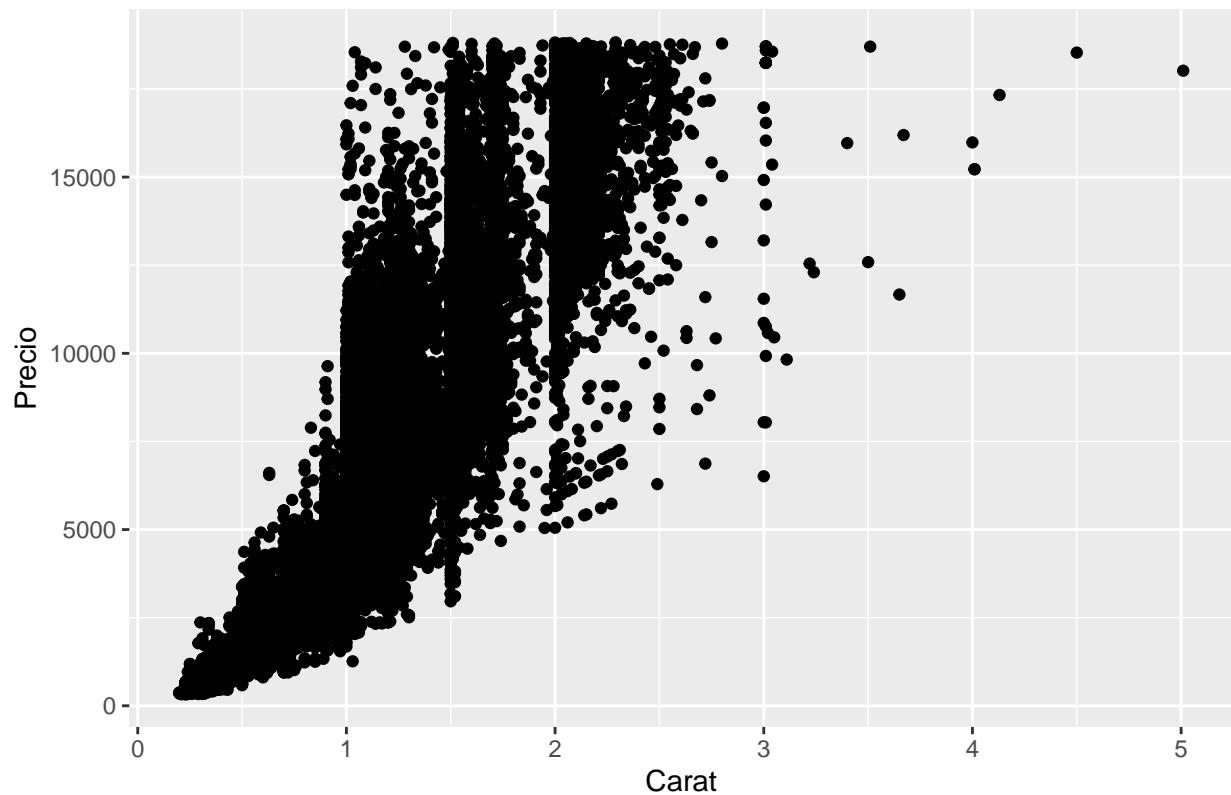


Vemos un resultado similar al obtenido para depth. Los valores de table se mueven principalmente en un rango comprendido entre 52 y 66 y para todos estos valores encontramos diamantes de todo el espectro de precios por lo que estas dos variables no presentan relación directa.

### CARAT (Quilates)

```
ggplot(diamonds, aes(x = carat, y = price)) +  
  geom_point() +  
  ggtitle("Relación entre el precio y el peso (carat)") +  
  ylab("Precio") +  
  xlab("Carat")
```

## Relación entre el precio y el peso (carat)



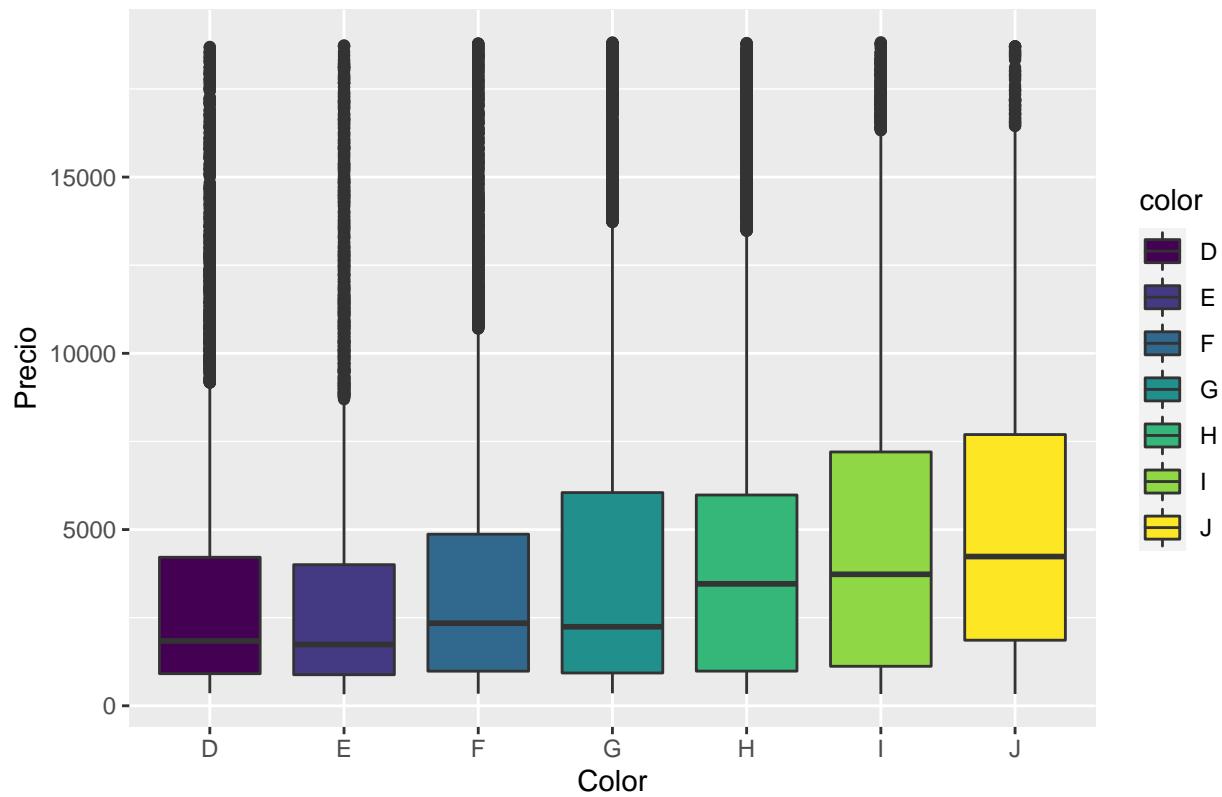
A pesar de la gran dispersión de los datos para esta variable si que se observa una clara tendencia a la alta, a más quilates tenemos un mayor precio.

## COLOR

La escala de color en los diamantes va desde la D hasta la J, de mejor a peor, respectivamente.

```
ggplot(diamonds, aes(x = color, y = price)) +  
  geom_boxplot(aes(fill = color)) +  
  ggtitle("Relación entre el precio y la calidad del color") +  
  ylab("Precio") +  
  xlab("Color")
```

## Relación entre el precio y la calidad del color



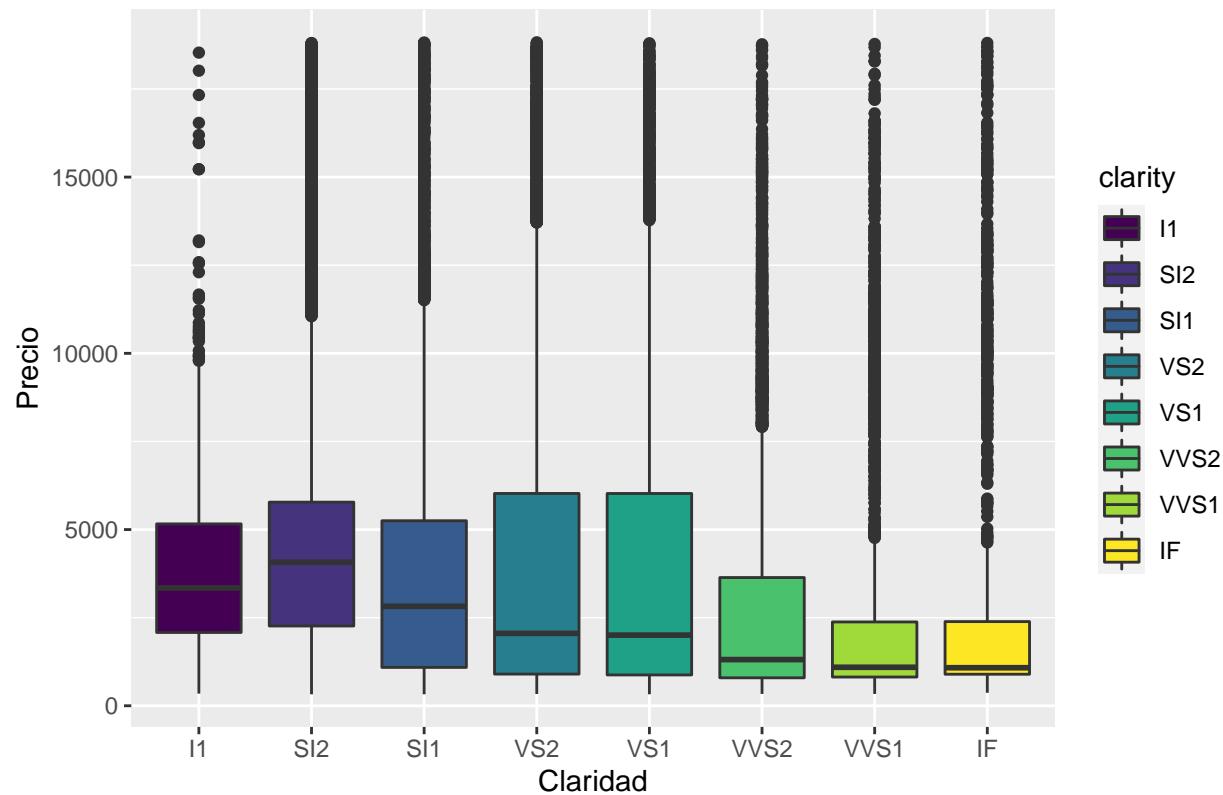
Vemos como existe una ligera relación entre estas dos variables, conforme peor es el color, mayor parece ser el precio medio de los diamantes pertenecientes a este color.

## CLARITY

La escala de la claridad varía desde I1 hasta IF, de peor a mejor, respectivamente.

```
ggplot(diamonds, aes(x = clarity, y = price)) +  
  geom_boxplot(aes(fill = clarity)) +  
  ggtitle("Relación entre el precio y la claridad del color") +  
  ylab("Precio") +  
  xlab("Claridad")
```

## Relación entre el precio y la claridad del color

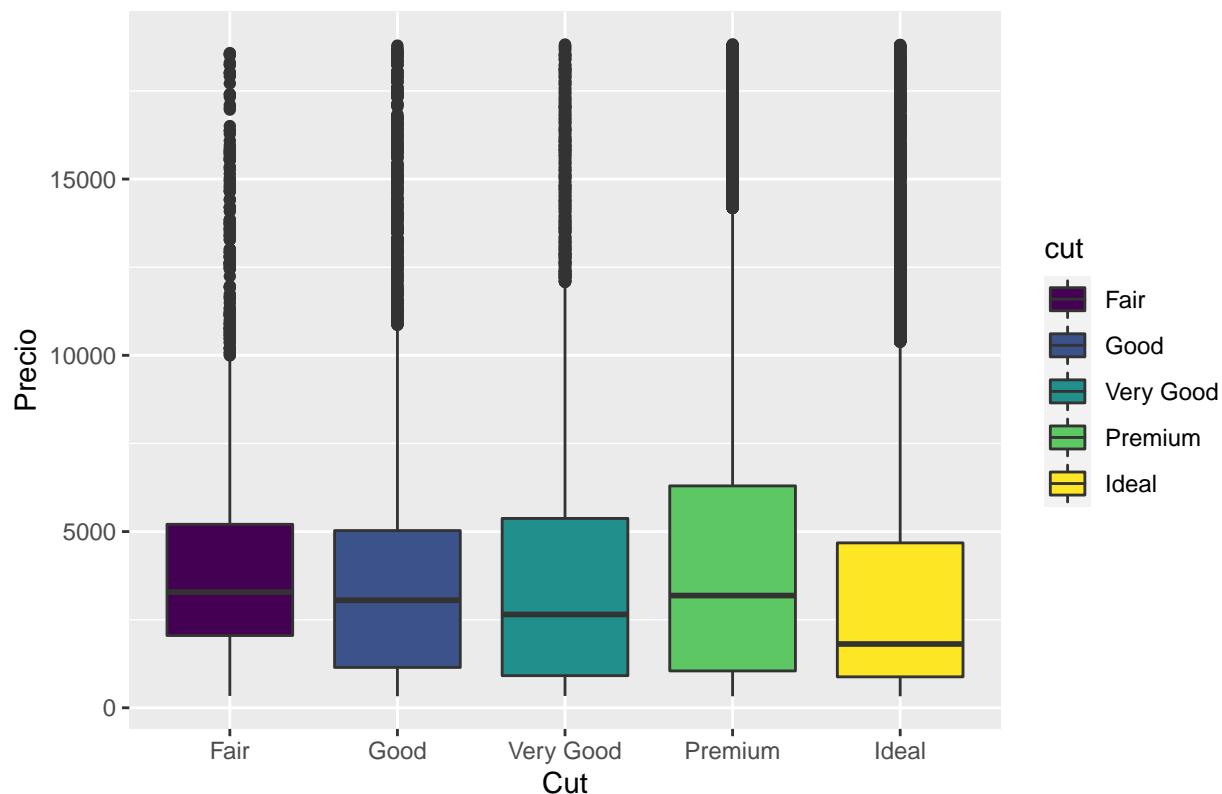


Para esta variable tambiñen observamos, al igual que con el color, una relaciñon ligera inversa pues los diamantes de mejor claridad son los de menor precio.

### CUT

```
ggplot(diamonds, aes(x = cut, y = price)) +
  geom_boxplot(aes(fill = cut)) +
  ggtitle("Relaciñon entre el precio y el tipo de corte (cut)") +
  ylab("Precio") +
  xlab("Cut")
```

## Relación entre el precio y el tipo de corte (cut)



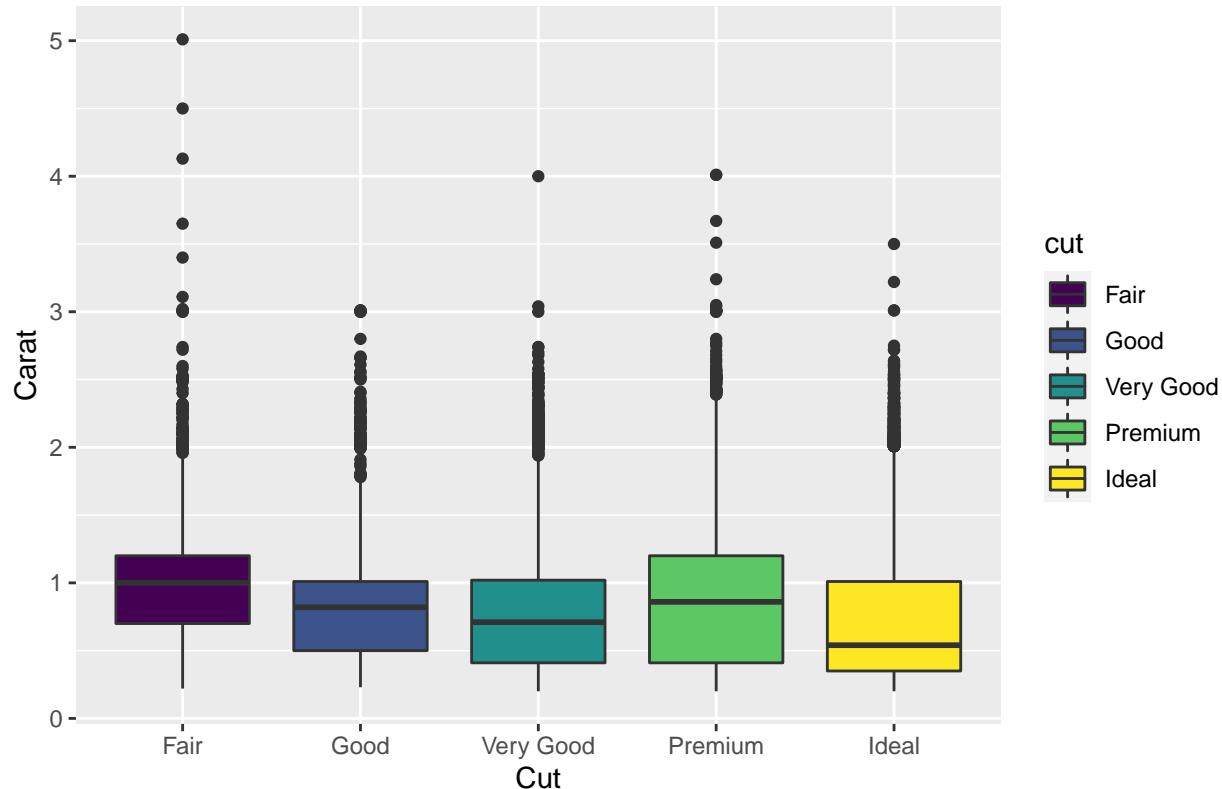
El tipo de corte no parece tener una relación estrecha con el precio en base a las medianas y las desviaciones observadas, ningún tipo de corte parece destacar en precio medio. Sin embargo, parece observarse de nuevo una ligera tendencia negativa teniendo un menor precio medio los diamantes con mejores cortes.

Finalmente tras haber hecho todas estas representaciones llegamos a la conclusión de que la variable que tiene un mayor peso y que está directamente relacionada con el precio de los diamantes es el peso de los mismos (carat), es decir, de cuántos quilates es un diamante.

Si analizamos la relación entre esta variable y la variable cut obtenemos lo siguiente:

```
ggplot(diamonds, aes(x = cut, y = carat)) +
  geom_boxplot(aes(fill = cut)) +
  ggtitle("Relación entre el peso y el tipo de corte") +
  ylab("Carat") +
  xlab("Cut")
```

## Relación entre el peso y el tipo de corte



Teniendo en cuenta la relación directa que habíamos observado entre el precio y la variable carat este gráfico es como cabría esperar, similar al que relacionaba cut con el precio. Vemos que existe una relación inversa, es decir, los diamantes que tienen un mejor corte presentan un peso menor de media. También se puede ver como que los diamantes más pesados suelen tener un corte peor, quizás porque su peso dificulta la tarea. Esto nos permite eplicar la relación inversa que observábamos entre precio y cut. Los diamantes con mejor corte suelen tener un precio menor de media, no porque el mejor corte abarate el diamante, sino porque los que tienen un mejor corte suelen ser menos pesados de media y, por tanto, tienden a valer menos.

Un posible análisis interesante podría ser analizar cómo varía el precio de los diamantes con el corte dentro de un grupo de diamantes del mismo peso, pues probablemente obtendríamos la relación contraria.

### Haz el ejercicio 4 de la Sección 12.6.1 de R4DS.

En este ejercicio se nos pide que:

Para cada país, año y sexo computa el número total de casos de tuberculosis. Haz una visualización informativa de los datos.

Primero aprovechamos el código previo de la sección para tener los datos limpios y facilitarnos el trabajo:

```
whoTidy <- (who %>%
  pivot_longer(
    cols = new_sp_m014:newrel_f65,
    names_to = "key",
    values_to = "cases",
    values_drop_na = TRUE
  ) %>%
  mutate(
    key = stringr::str_replace(key, "newrel", "new_rel")
```

```

) %>%
separate(key, c("new", "var", "sexage")) %>%
select(-new, -iso2, -iso3) %>%
separate(sexage, c("sex", "age"), sep = 1))

head(whoTidy)

## # A tibble: 6 x 6
##   country     year var   sex   age cases
##   <chr>      <int> <chr> <chr> <chr> <int>
## 1 Afghanistan 1997 sp    m    014     0
## 2 Afghanistan 1997 sp    m    1524    10
## 3 Afghanistan 1997 sp    m    2534     6
## 4 Afghanistan 1997 sp    m    3544     3
## 5 Afghanistan 1997 sp    m    4554     5
## 6 Afghanistan 1997 sp    m    5564     2

```

Para contar el número de casos totales anuales en cada país en función del sexo deberemos hacer un group\_by country, year y sex para que sume los casos diferenciando solo estas variables:

```

whoReduc <- (whoTidy %>%
  group_by(country, year, sex) %>%
  summarise(totalCases = sum(cases)))

```

```

## `summarise()` has grouped output by 'country', 'year'. You can override using the `.groups` argument
head(whoReduc)

```

```

## # A tibble: 6 x 4
## # Groups:   country, year [3]
##   country     year sex totalCases
##   <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

```

Realizaremos una primera representación de todos los datos para tener una idea de lo que podemos mejorar o qué cosas pueden resultar más interesantes. Para graficar uniremos las columnas correspondientes al sexo y al país para poder agrupar los casos de los distintos años a esta variable

```

whoReduc <- (whoReduc %>%
  unite(countrySex, country, sex, remove = FALSE))

head(whoReduc)

```

```

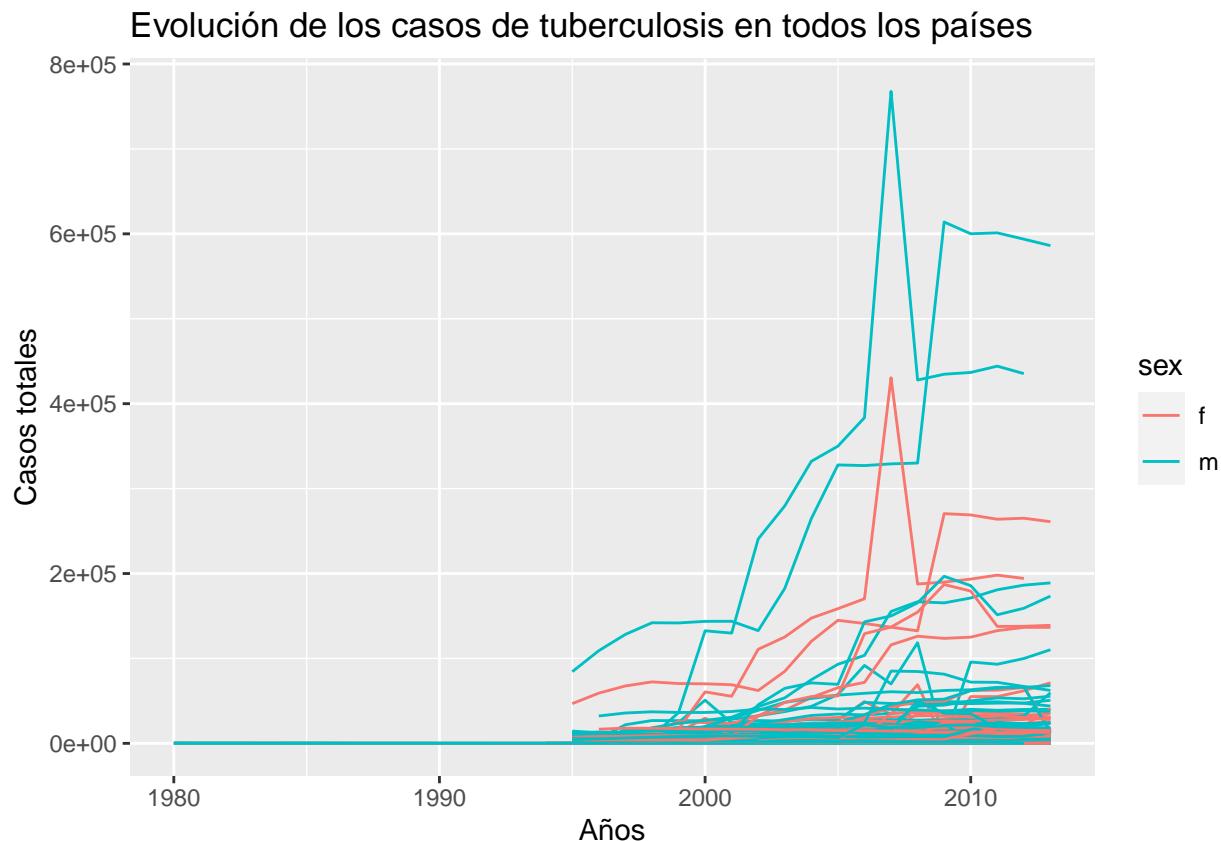
## # A tibble: 6 x 5
## # Groups:   country, year [3]
##   countrySex   country     year sex   totalCases
##   <chr>        <chr>      <int> <chr>     <int>
## 1 Afghanistan_f Afghanistan 1997 f        102
## 2 Afghanistan_m Afghanistan 1997 m        26
## 3 Afghanistan_f Afghanistan 1998 f       1207
## 4 Afghanistan_m Afghanistan 1998 m        571
## 5 Afghanistan_f Afghanistan 1999 f        517

```

```
## 6 Afghanistan_m Afghanistan 1999 m 228
```

Ahora graficamos:

```
ggplot(whoReduc, aes(x = year, y = totalCases, group = countrySex, colour = sex)) +  
  geom_line() +  
  ggtitle("Evolución de los casos de tuberculosis en todos los países") +  
  ylab("Casos totales") +  
  xlab("Años")
```



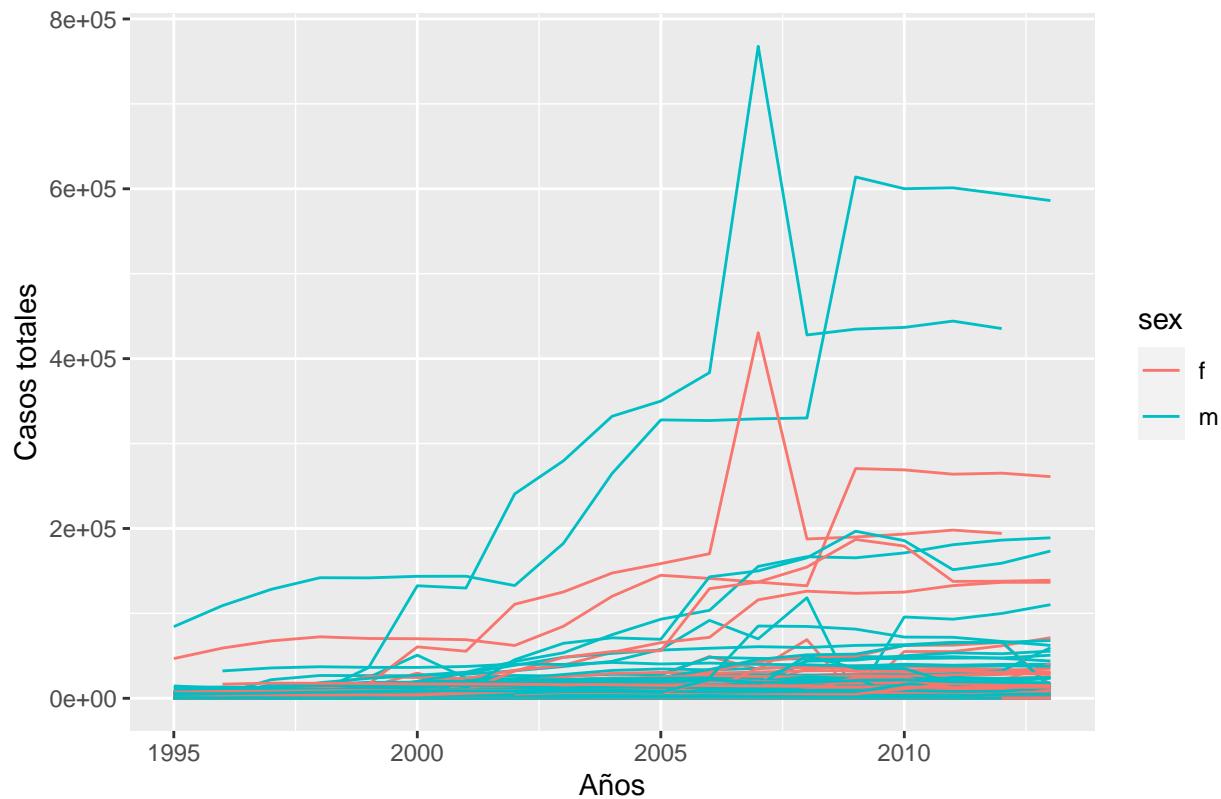
Vemos que hasta 1995 no hay datos disponibles que nos proporcionen información, por lo que eliminaremos estos registros de la tabla seleccionando únicamente aquellos que nos interesan:

```
whoReduc <- (whoReduc %>%  
  filter(year >= 1995))
```

Volvemos a realizar el gráfico con nuestros datos filtrados para verlo mejor:

```
ggplot(whoReduc, aes(x = year, y = totalCases, group = countrySex, colour = sex)) +  
  geom_line() +  
  ggtitle("Evolución de los casos de tuberculosis en todos los países") +  
  ylab("Casos totales") +  
  xlab("Años")
```

## Evolución de los casos de tuberculosis en todos los países



La visualización de los datos sigue siendo deficiente, al haber asignado el color en función del sexo no podemos distinguir unos países de otros. Además la enorme cantidad de líneas para casos bajos impide ver estos datos con claridad.

Lo que haremos será graficar por separado los países que tengan un mayor número de casos. Para ello obtenemos los países que han tenido un mayor número de casos en total desde 1995:

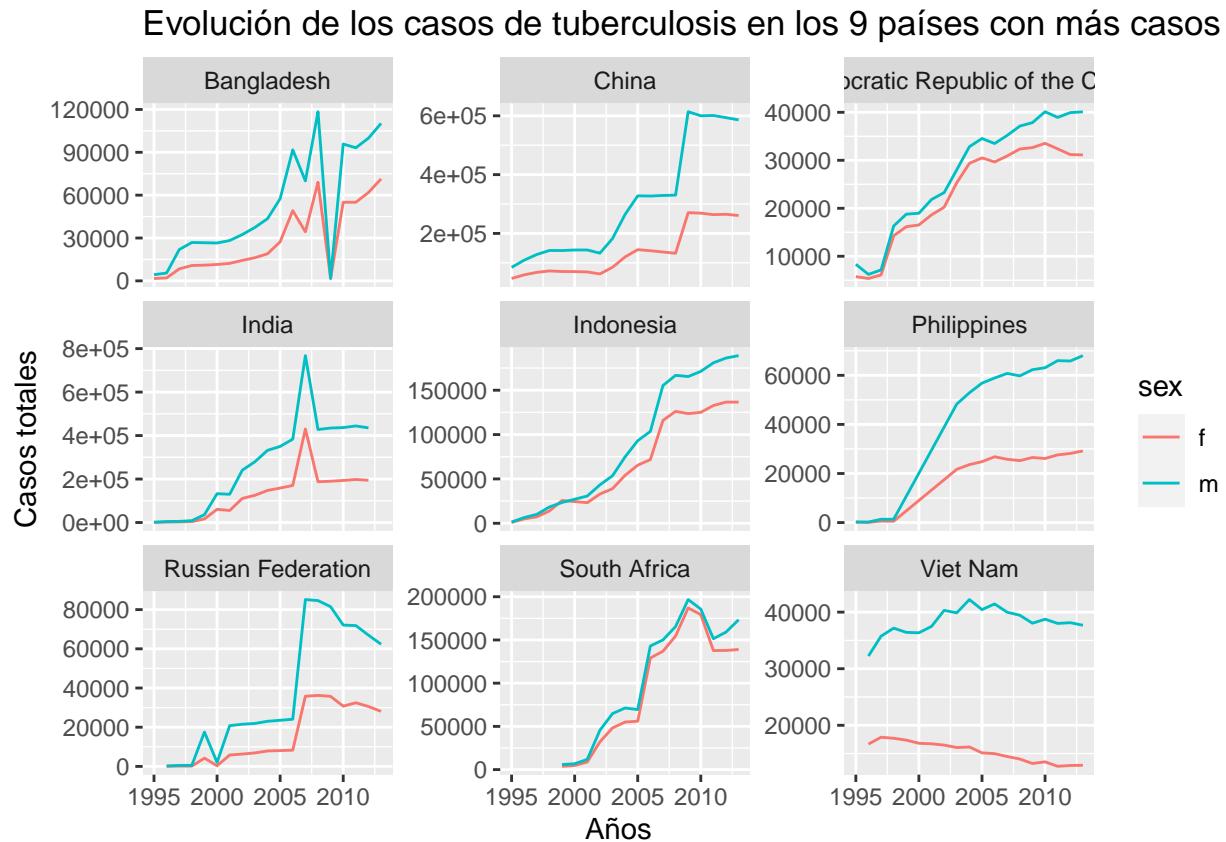
```
whoReducTop <- (whoReduc %>%
  group_by(country) %>%
  summarise(casosHistoricos=sum(totalCases)) %>%
  arrange(desc(casosHistoricos)) %>%
  head(9))
```

whoReducTop

```
## # A tibble: 9 x 2
##   country           casosHistoricos
##   <chr>                  <int>
## 1 China                8389839
## 2 India                 7098552
## 3 South Africa          3010272
## 4 Indonesia              2909925
## 5 Bangladesh             1524034
## 6 Viet Nam               965665
## 7 Democratic Republic of the Congo 960902
## 8 Philippines            952828
## 9 Russian Federation     926236
```

Vemos que la lista la encabeza China, seguida de India. Ahora graficaremos los casos en función del sexo para estos países por separado:

```
whoReduc %>%
  filter(country %in% whoReducTop$country) %>%
  ggplot(aes(x = year, y = totalCases, group = countrySex, colour = sex)) +
  geom_line() +
  facet_wrap(~ country, scales = 'free_y') +
  ggtitle("Evolución de los casos de tuberculosis en los 9 países con más casos") +
  ylab("Casos totales") +
  xlab("Años")
```



Una vez graficadas las evoluciones de los casos en estos 9 países podemos extraer diversas conclusiones. Para empezar se observa que la mayoría de estos países son países del tercer mundo o en vías de desarrollo, que generalmente presentan peores sistemas de salud que los demás países, lo cuál puede explicar el elevado número de casos. Por otro lado, tenemos que siempre se observan más casos en los hombres que en las mujeres. Esto tiene sentido si tenemos en cuenta que en la mayoría de estos países sigue existiendo una diferencia grande entre sexos, desempeñando los hombres un papel más relacionado con el trabajo y, por tanto, el contacto con otras personas, quedando las mujeres relegadas a un papel más de amas de casa que supone un menor riesgo de contagio. Otro factor a destacar es que las curvas de los hombres y de las mujeres se comportan de manera similar dentro de cada país, es decir, ante un aumento de casos este suele suceder tanto en hombres como en mujeres al mismo tiempo. Los datos muestran en la mayoría de países una tendencia a la alta. Esta podría explicarse teniendo en cuenta un posible aumento en los sistemas de detección y diagnóstico en los últimos años o un incremento en la población que conllevaría un aumento en el número total de casos.