

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

Tarea 2

Carlos García Vázquez

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

Instrucciones preliminares

- Empieza abriendo el proyecto de RStudio correspondiente a tu repositorio personal de la asignatura.
- En todas las tareas tendrás que repetir un proceso como el descrito en la sección *Repite los pasos Creando un fichero Rmarkdown para esta práctica* de la *Práctica00*. Puedes releer la sección *Practicando la entrega de las Tareas* de esa misma práctica para recordar el procedimiento de entrega.

Preliminares

Carga de librerías inicial:

```
library(tidyverse)
```

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 $P(X = x_i)$	$\frac{64}{125}$	$\frac{48}{125}$	$\frac{12}{125}$	$\frac{1}{125}$

Creamos los vectores para trabajar con ellos.

```
X1=c(0:3)
pX1=c(64/125,48/125,12/125,1/125)
(tableX1=data.frame(X1,pX1))
```

```
##   X1     pX1
## 1  0  0.512
## 2  1  0.384
## 3  2  0.096
## 4  3  0.008
```

Calcula la media y la varianza teóricas de esta variable.

```
#Media teórica  
(mu_t=sum(X1*pX1))  
  
## [1] 0.6  
  
#Varianza teórica  
(var_t=sum(((X1-mu_t)^2)*pX1))  
  
## [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.

Estudiaremos la distribución de las medias muestrales en 2 casos, entre los cuales, la principal diferencia será el tamaño de la muestra (n) empleado.

Número de muestras para ambos casos ($k=100000$)

```
set.seed(2021)  
k=100000
```

ANÁLISIS 1 ($n=10$)

1) Creamos el vector con las medias muestrales de las 100000 muestras generadas

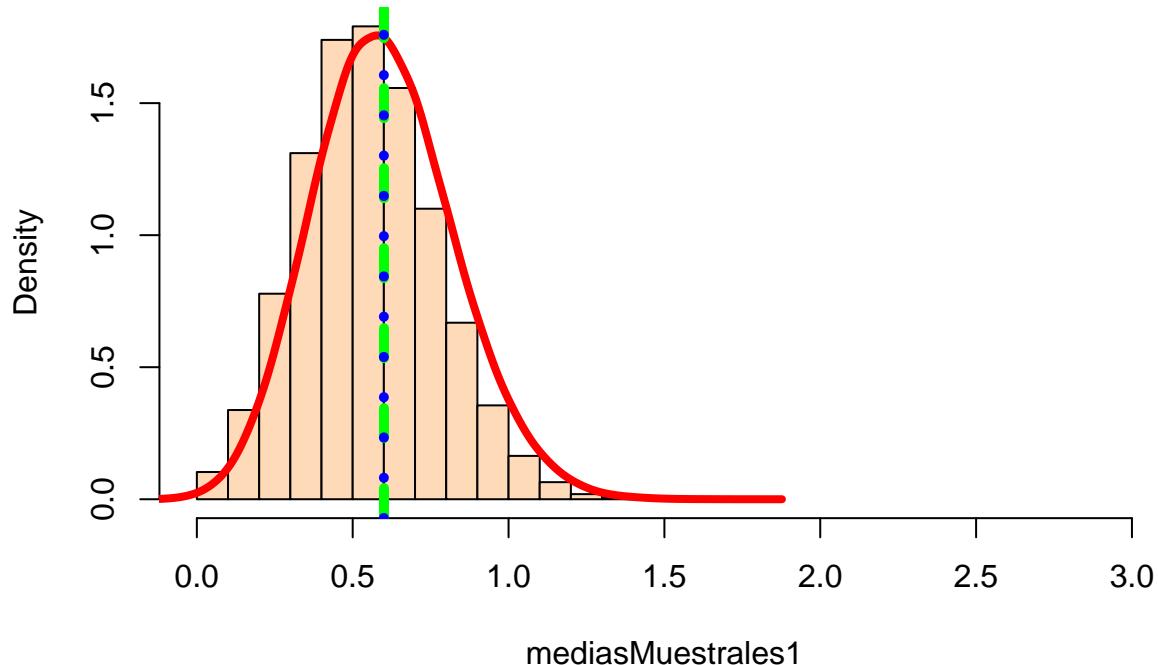
```
n1=10  
mediasMuestrales1 = replicate(k, {  
  muestra = sample(X1, n1, replace = TRUE, prob=pX1)  
  mean(muestra)  
})
```

2) Estudio de la distribución de las medias muestrales

A continuación, podemos apreciar la representación gráfica de la distribución de las medias muestrales

```
hist(mediasMuestrales1, breaks = 20, main="Distribución de medias muestrales (n=10)",  
  col="peachpuff", probability = TRUE, xlim=range(X1))  
lines(density(mediasMuestrales1,adjust=3), lwd=4, col="red")  
abline(v = mean(mediasMuestrales1), lty=2, lwd=5, col="green")  
abline(v = mu_t, lty=3, lwd=5, col="blue")
```

Distribución de medias muestrales (n=10)



En cuanto a la distribución, en primer lugar se puede apreciar que es asimétrica a la derecha. Aparte, las líneas discontinuas en color verde y azul ubicadas en el gráfico, representan las media obtenida a partir de las muestras tomadas y la media teórica, respectivamente. En este aspecto, destacar el hecho de que estos 2 valores coinciden, por lo que se podría decir que a partir de las 100000 muestras de tamaño 10 tomadas, se ha conseguido una buena estimación de la media teórica de la población.

Ahora, lo que vamos a hacer es aumentar el tamaño de las muestras tomadas.

ANÁLISIS 2 (n=30)

- 1) Creamos el vector con las medias muestrales de las 100000 muestras generadas

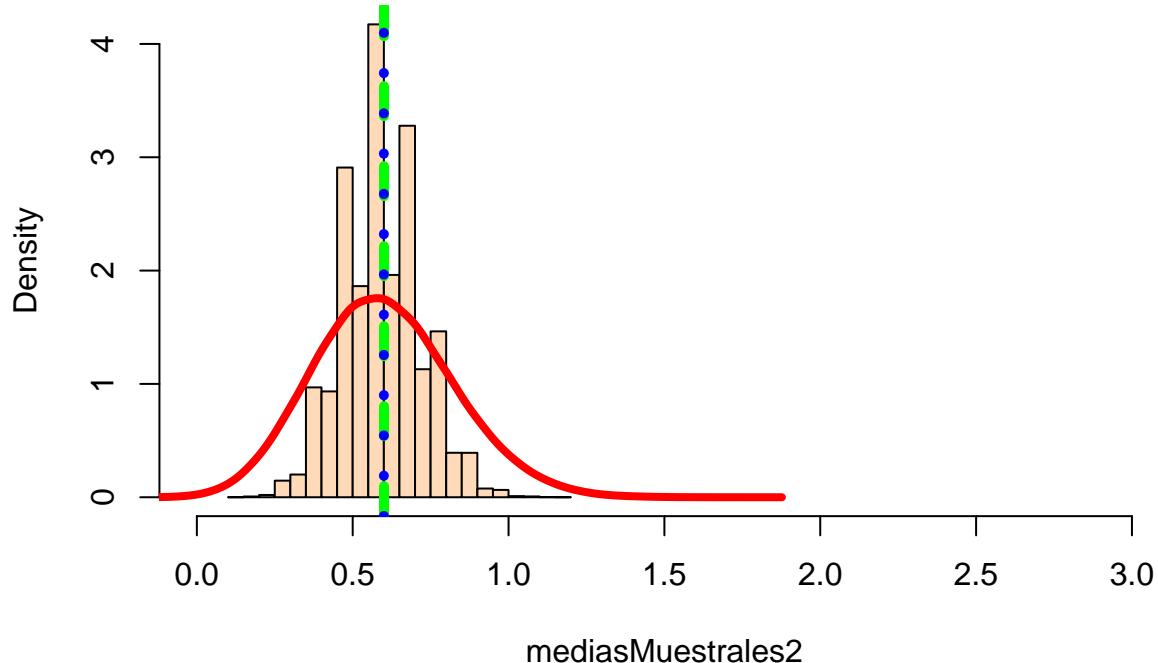
```
n2=30  
mediasMuestrales2 = replicate(k, {  
  muestra = sample(X1, n2, replace = TRUE, prob=pX1)  
  mean(muestra)  
})
```

- 2) Estudio de la distribución de las medias muestrales

Representación gráfica de la distribución de las medias muestrales

```
hist(mediasMuestrales2, breaks = 20, main="Distribución de medias muestrales (n=30)",  
  col="peachpuff", probability = TRUE, xlim=range(X1))  
lines(density(mediasMuestrales1,adjust=3), lwd=4, col="red")  
abline(v = mean(mediasMuestrales2), lty=2, lwd=5, col="green")  
abline(v = mu_t, lty=3, lwd=5, col="blue")
```

Distribución de medias muestrales (n=30)



Aquí, al igual que en el caso anterior, se aprecia la asimetría a la derecha, y la media obtenida a partir de las muestras y la teórica, que tambien coinciden en el gráfico.

CONCLUSIÓN

En ambos casos, hemos visto que la media de las medias muestrales coincide con la media de la población.

Aparte, se confirma lo que dice el conocido como Teorema Central del Límite, que afirma, que dado un tamaño de muestra aleatoria de la población lo suficientemente grande, la distribución de las medias muestrales seguirá una distribución normal, tal y como hemos podido comprobar en ambos casos.

Otra de las cosas a destacar en la comparación entre ambos gráficos, es la reducción de la anchura de la campana entorno a la media, al aumentar el tamaño de las muestras de 10 a 30. Por lo que la variabilidad en cuanto a las medias de las muestras disminuye, cumpliendo tambien con las afirmaciones del Teorema Central del Límite. En términos generales, a mayor tamaño de muestra, la precisión con la que obtenemos la media será mayor.

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?

Nuestra variable a estudiar es $X_1 + X_2$

Tabla de probabilidades de X_2

```
X2=c(0:2)
pX2=c(1/2,1/4,1/4)
(tableX2=data.frame(X2,pX2))
```

```
##   X2  pX2
## 1  0  0.50
## 2  1  0.25
## 3  2  0.25
```

```
#Media teórica
(mu_t1=sum(X2*pX2))
```

```
## [1] 0.75
```

```
#Varianza teórica
(var_t1=sum(((X2-mu_t1)^2)*pX2))
```

```
## [1] 0.6875
```

A continuación, vamos a crear una tabla con las distintas posibilidades que se pueden dar entre ambos conjuntos de datos, cuento sería la suma entre ambos y la probabilidad de que sucedan, teniendo en cuenta las tablas de probabilidad de ambas variables.

Al ser variables independientes X1 y X2, la intersección de los sucesos se calcula como la multiplicación de las probabilidades de cada uno.

```
pos=merge(tableX1$X1,tableX2$X2) %>%
  mutate(posSum=x+y,prob=rep(tableX1$pX1,times=3)*rep(tableX2$pX2,each=4))
rename(pos,X1=x,X2=y)
```

```
##   X1  X2 posSum  prob
## 1  0   0      0  0.256
## 2  1   0      1  0.192
## 3  2   0      2  0.048
## 4  3   0      3  0.004
## 5  0   1      1  0.128
## 6  1   1      2  0.096
## 7  2   1      3  0.024
## 8  3   1      4  0.002
## 9  0   2      2  0.128
## 10 1   2      3  0.096
## 11 2   2      4  0.024
## 12 3   2      5  0.002
```

Convertimos la columna ‘posSum’ en factor para hacer la clasificación y agrupar por cada resultado posible

```
pos$posSum=as.factor(pos$posSum)
```

Realizamos el cálculo final de la tabla de probabilidades de la nueva variable X1 + X2

```
pos %>%
  group_by(posSum) %>%
  summarise(prob=sum(prob))
```

```
## # A tibble: 6 x 2
##   posSum   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.

Cálculo de la media teórica de la variable X1+X2

- 1) Cálculo a partir de los resultados obtenidos en el apartado anterior

Tener en cuenta que con el `as.numeric` y la función `paste`, conseguimos transformar los valores reales de la columna `posSum` del DataFrame `pos` de tipo factor a numérico.

```
#Media teórica
(mu_t2=sum(as.numeric(paste(pos$posSum))*pos$prob))
```

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

- 2) Cálculo a partir de las medias de cada variable

La media de la suma es la suma de las medias

$$E(X_1 + X_2) = E(X_1) + E(X_2)$$

```
(mu_suma=mu_t+mu_t1)
```

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

Hemos calculado la media teórica de las 2 formas, y como era evidente el resultado es el mismo.

En este caso, cogemos `k=100000` muestras de tamaño 1 cada una de ambas variables para estudiar la suma de las mismas

```
set.seed(2021)
k=100000
n3=1
```

Calculamos las medias muestrales de la variable `X1+X2`, sabiendo que su media será igual a la suma de la media de `X1` más la media de `X2`:

```

mediasMuestrales3 = replicate(k, {
  muestraX1 = sample(X1, n3, replace = TRUE, prob=pX1)
  muestraX2 = sample(X2, n3, replace = TRUE, prob=pX2)
  muestraX1 + muestraX2
})

```

La media de estos valores es:

```
(mediaFinal=mean(mediasMuestrales3))
```

```
## [1] 1.35143
```

Podemos ver que sale igual, a excepción de algunos decimales, que la media teórica calculada previamente en el apartado anterior.

En este caso hemos reducido el tamaño de la muestra lo máximo hasta tomar una única, haciendo que la variabilidad de las medias obtenidas en las muestras sea mayor, y reduciendo de esta forma la precisión en cuanto a la estimación de la media con respecto a la teórica.

Ejercicio 2. Datos limpios

- Descarga el fichero de este enlace

<https://gist.github.com/fernandosansegundo/471b4887737cfcec7e9cf28631f2e21e/raw/b3944599d02df494f5903740>
testResults.csv

Una vez descargado el fichero, lo cargamos, vemos su estructura y como está organizado

```
testResults=read_csv("./data/testResults.csv")
```

```
## Rows: 200 Columns: 9
```

```

## -- Column specification --
## Delimiter: ","
## chr (2): name, gender_age
## dbl (7): id, test_number, week1, week2, week3, week4, week5

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```

```
head(testResults)
```

```

## # 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

```

```

names(testResults)

## [1] "name"         "id"           "gender_age"    "test_number"   "week1"
## [6] "week2"        "week3"        "week4"        "week5"

```

- 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*.

- 1) Explicación de por qué no cumple con los principios de tidy data

Este conjunto de datos que se está estudiando, no es limpio, porque las filas no corresponden con observaciones y las columnas no corresponden con variables, que son precisamente 2 de las 3 condiciones para que un conjunto de datos se considere limpio.

- 2) Obtención de tabla de datos limpios con la misma información

Primero, hay que convertir los valores de semana en una columna, lo que hará la tabla más larga y estrecha. Todo esto haciendo uso de la función ‘pivot_longer()’

Luego, sepáramos la variable `gender_age` en las variables `gender` y `age` correspondientes, haciendo uso de la función `separate()`

Separamos la columna `week` para quedarnos con el valor numérico que identifica a cada una de las semanas, y borramos la columna en la que se queda el valor repetido ‘`week`’

Para terminar y dejar el DataFrame limpio, pasamos la columna `week` de tipo ‘character’ a numérica. Aunque luego, en función del estudio que se pretenda realizar sobre el conjunto de datos en cuestión, se podrá cambiar el tipo de variable, según interese.

```

ResultsTidy = testResults %>%
  pivot_longer(week1:week5, names_to = "week") %>%
  separate(gender_age,c("gender","age"),sep="_") %>%
  separate(week,c('w','week'),sep=4)%>%
  mutate(w=NULL)
ResultsTidy$week=as.numeric(ResultsTidy$week)
ResultsTidy

```

```

## # A tibble: 1,000 x 7
##   name     id gender age  test_number  week value
##   <chr>   <dbl> <chr> <chr>      <dbl> <dbl> <dbl>
## 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
## 7 Jacob    108 m    20          2       2     2
## 8 Jacob    108 m    20          2       3     4
## 9 Jacob    108 m    20          2       4     0
## 10 Jacob   108 m    20          2       5     3
## # ... with 990 more rows

```

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.

- Haz el ejercicio 2 de la Sección 7.5.1.1 de R4DS. Las ideas de esa sección son importantes para nuestro trabajo de las próximas sesiones.

Analizamos la estructura del DataFrame y las variables que forman parte del mismo

```
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat     cut      color clarity depth table price     x     y     z
##   <dbl> <ord>    <ord> <ord>   <dbl> <dbl> <int> <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
```

```
names(diamonds)
```

```
## [1] "carat"    "cut"       "color"     "clarity"   "depth"    "table"    "price"
## [8] "x"         "y"         "z"
```

What variable in the diamonds dataset is most important for predicting the price of a diamond? How is that variable correlated with cut? Why does the combination of those two relationships lead to lower quality diamonds being more expensive?

Para tener una primera información acerca del dataset y lo que representan las variables que contiene, accedemos a la ayuda de ‘diamonds’

- 1) Cuál es la variable del dataset más importante para la predicción de la variable precio de un diamante?

Tenemos que estudiar todas las variables y su relación e influencia sobre el precio de un diamante. Tras una primera visualización de todas las variables, nos damos cuenta de que los valores de x,y,z ya están incluidos o se tienen en cuenta en aquellas variables que estudian las dimensiones del diamante o que se ven influidas por estas, como podría ser ‘depth’.

Por lo tanto las variables a estudiar son :carat(quilates),cut(corte),color,clarity,depth y table

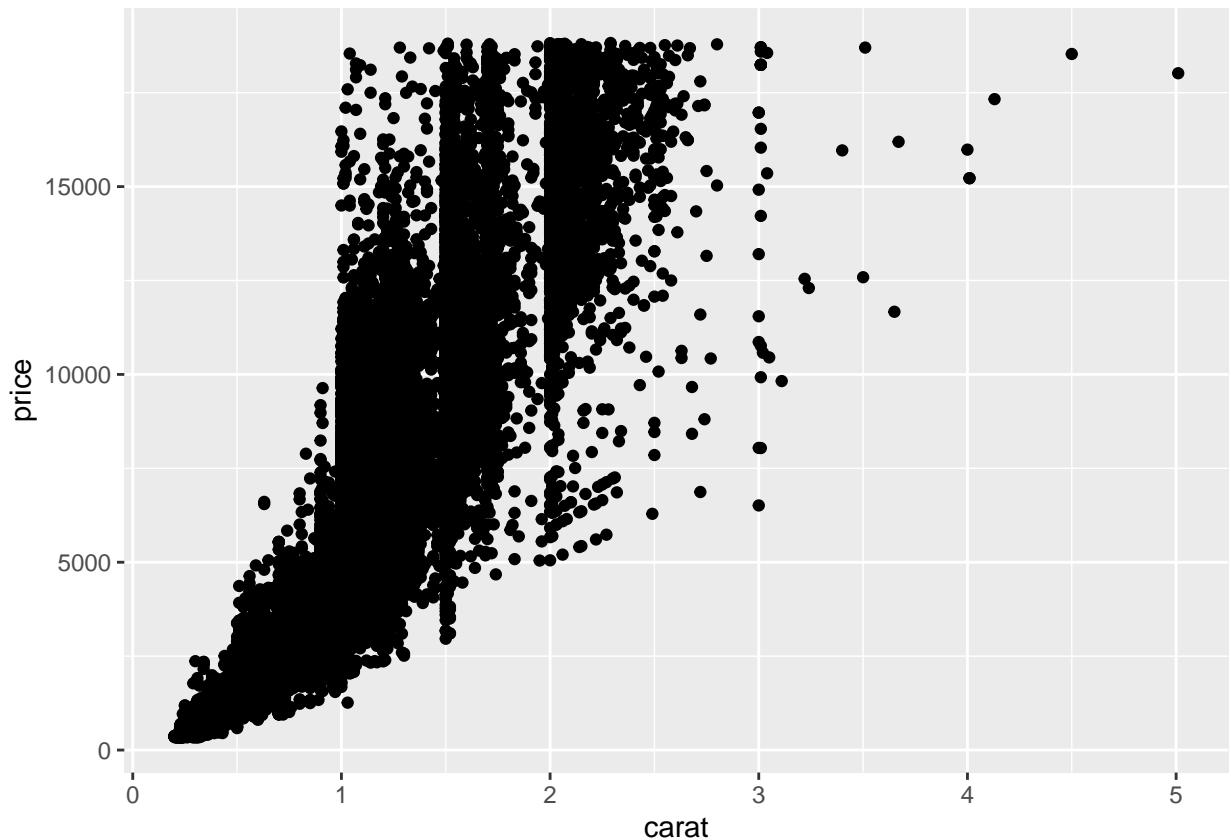
En términos generales, se va a llevar a cabo un estudio bidireccional en el que iremos comparando la relación de todas las variables mencionadas con la variable de estudio (price)

En función de la tipología de variable con la que estemos comparando la variable precio, realizaremos diagramas de dispersión o Boxplots. Concretamente, para las variables continuas emplearemos el diagrama de dispersión y para las categóricas el boxplot.

Primero realizaremos los diagramas de dispersión para las variables continuas, y posteriormente pasaremos a las categóricas con los boxplots, para realizar el estudio de forma ordenada.

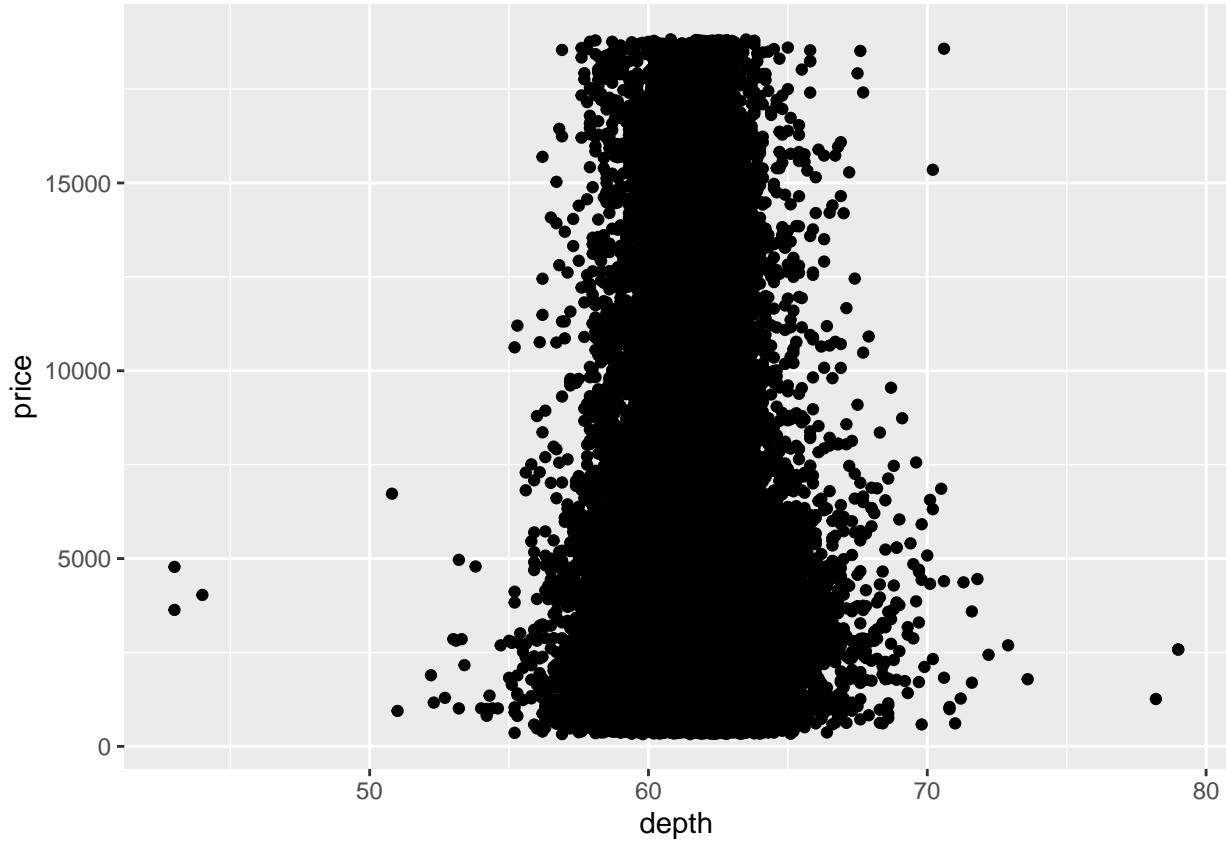
VARIABLES CONTINUAS (Diagrama de dispersión):

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



Aquí, a pesar de que la variabilidad del precio con respecto a los quilates es notable, se aprecia cierta tendencia. Concretamente, se ve que a medida que aumenta el peso, el precio también aumenta, por lo que se aprecia cierta relación entre ambas, lo que posiciona a la variable carat como candidata importante a considerar por la importancia que puede llegar a tener en la predicción del precio de un diamante.

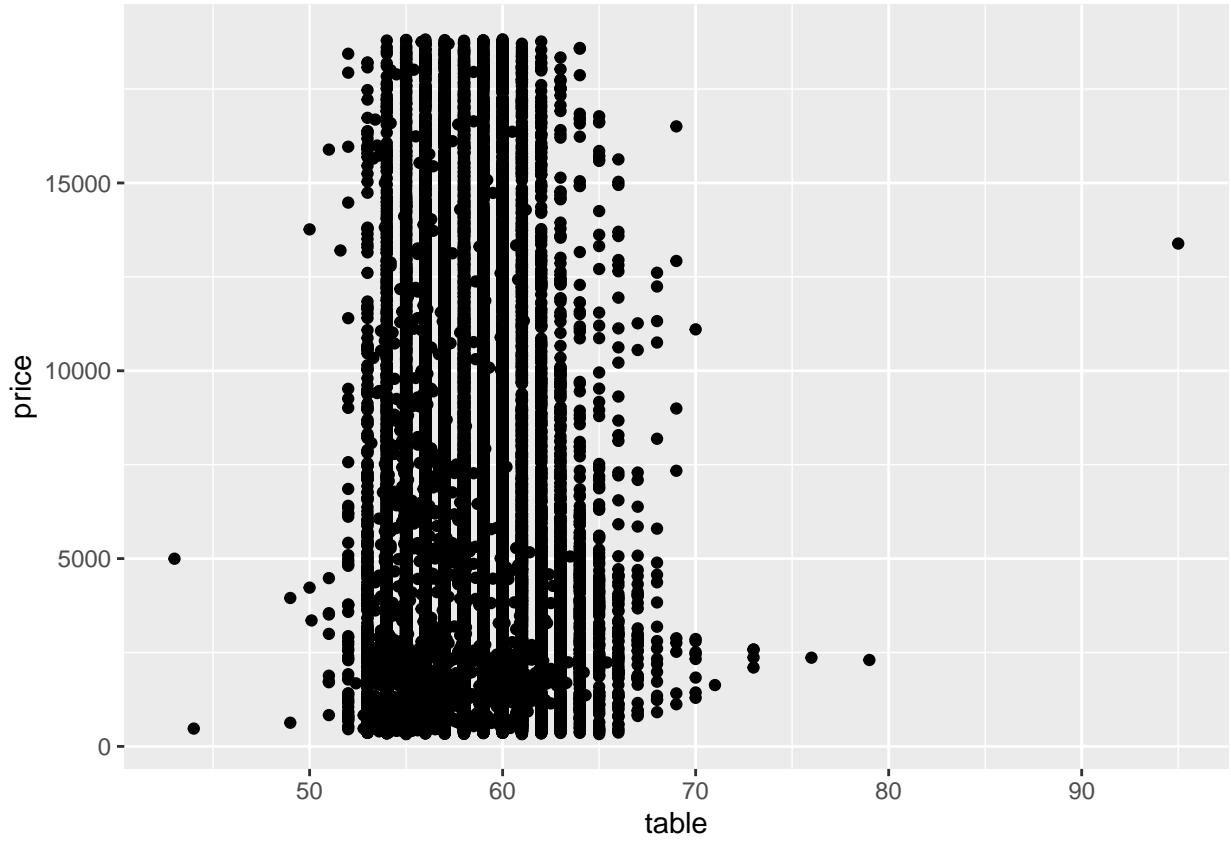
```
ggplot(diamonds, aes(x = depth, y = price)) +  
  geom_point()
```



En cuanto a la variable depth, se puede ver en el gráfico como hay una concentración de puntos entre 50 y 70 mm de profundidad, en el que la variabilidad de precios es muy alta, por lo que no se ve una relación clara entre ambas variables.

La variable table la tratamos como variable continua por el rango de valores que oscila.

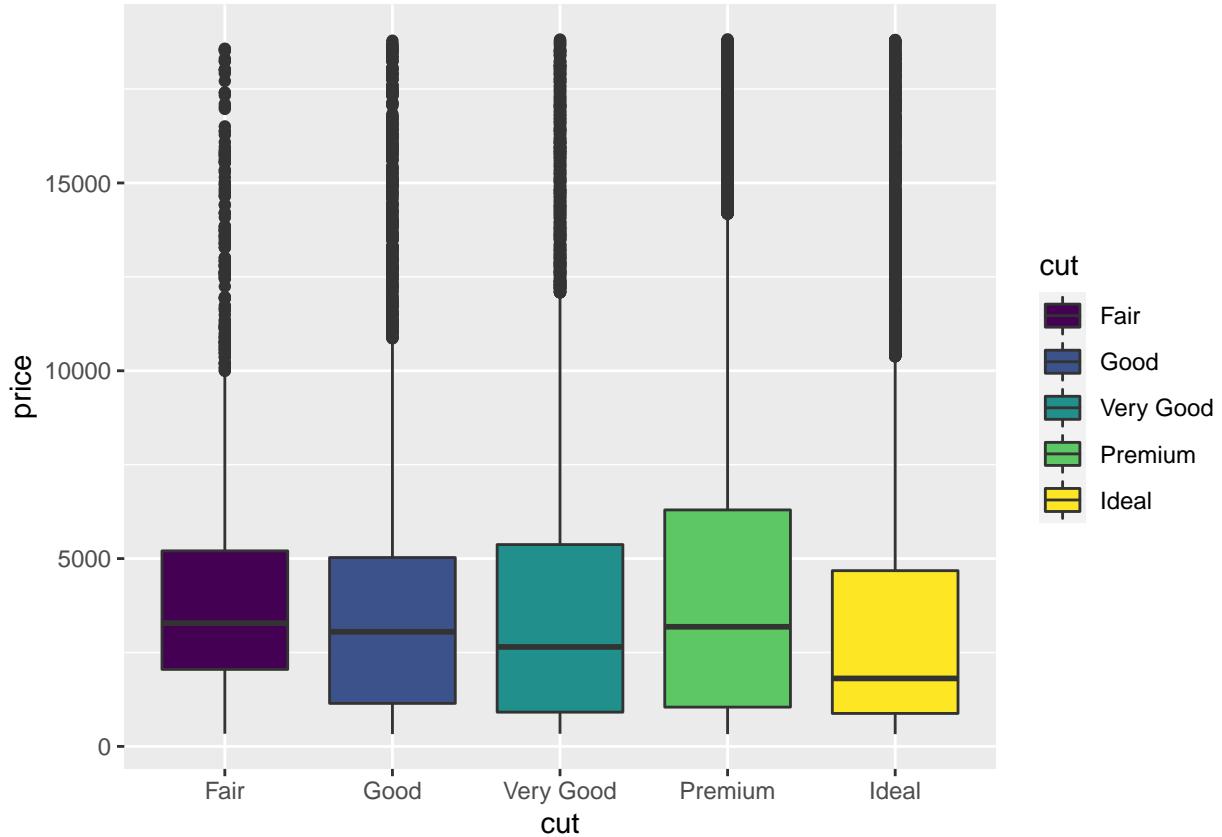
```
ggplot(diamonds, aes(x = table, y = price)) +  
  geom_point()
```



En este último gráfico de dispersión, no se ve ninguna tendencia ni indicio que sugiera que existe una relación entre la variable table y price. El caso, en cierta forma, es similar al de depth, con una variabilidad de precios muy alta para cada valor de table.

VARIABLES CATEGÓRICAS (Boxplot):

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

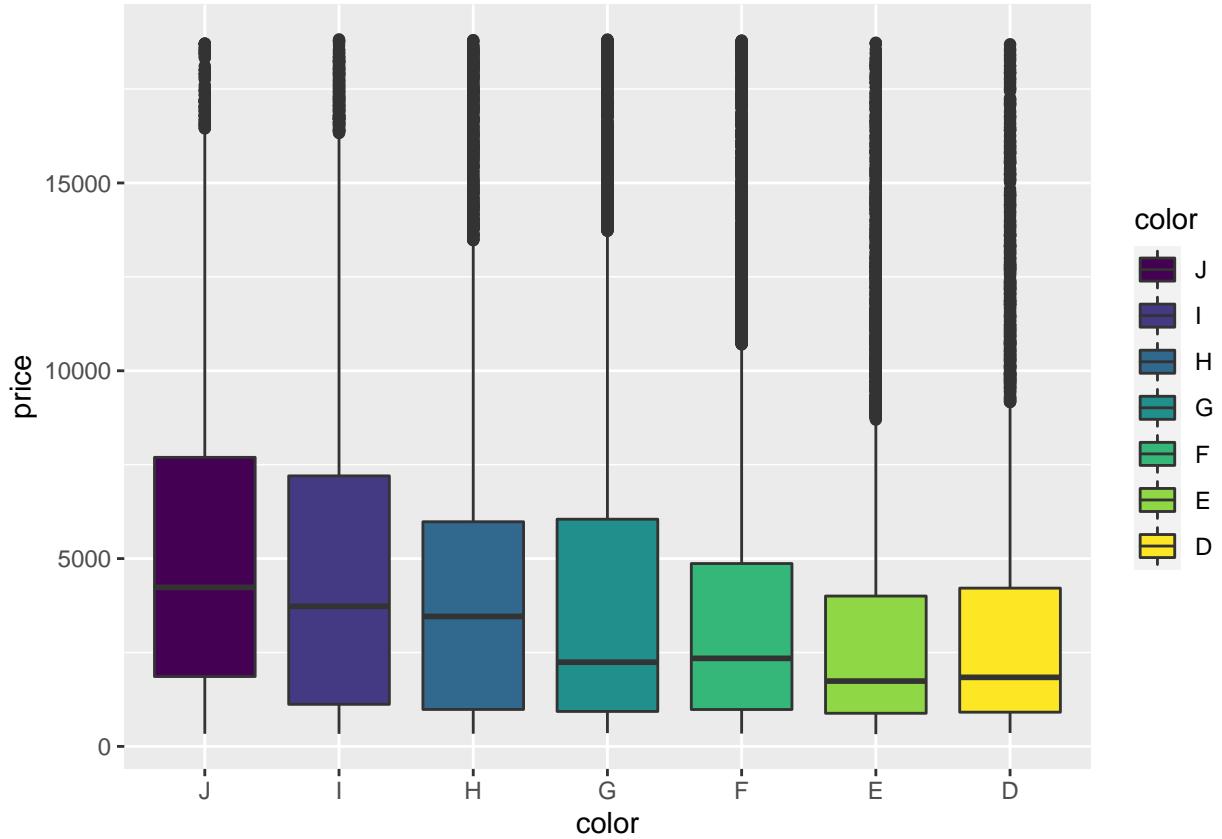


En este caso, tenemos la variable categórica ‘cut’, que ya se encuentra ordenada en función de la calidad del corte.

Al estudiar el gráfico, nos damos cuenta de que el rango de precios no varía demasiado en función del corte, ya que la variabilidad entre los niveles de la variable cut es reducida. Debido a esto, a priori, la variable cut no será considerada como un factor influyente en la predicción del precio del diamante.

Sí que es verdad, que conviene destacar que el corte ideal, que pensabamos que sería el más caro, tiene la mediana por debajo del resto, y habrá que descubrir el porqué.

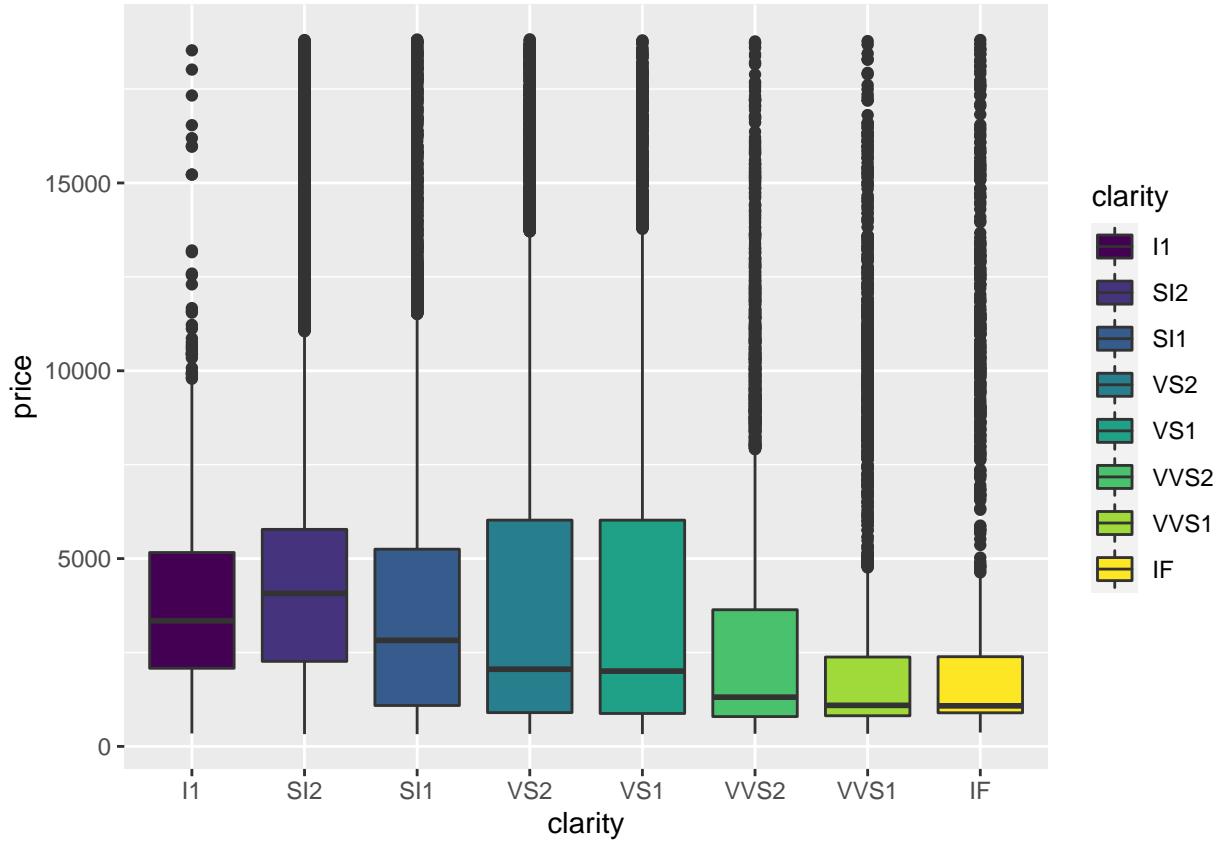
```
diamonds %>%
  mutate(color = fct_rev(color)) %>%
  ggplot(aes(x = color, y = price)) +
  geom_boxplot(aes(fill=color))
```



En cuanto a este caso, primero decir que se utiliza la función `fct_rev()` para ordenar los niveles de la variable color de forma ascendente a lo largo del eje X, y de esta forma poder analizar si es un factor influyente o no en el establecimiento del precio del diamante.

Se ve que cuanto peor es el color, la variabilidad aumenta y el rango de precios q se oscila parece mayor, lo que es algo confuso a priori. Por este motivo, se puede concluir que existe una relación negativa débil entre estas variables, aunque nada demasiado relevante para el análisis.

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



En esta última variable que se va a estudiar, los niveles de claridad también se encuentran ordenados de peor(I1) a mejor(IF), por lo que directamente procedemos a la realización de la gráfica.

En este caso, también se aprecia una débil relación negativa entre ambas variables, pero tampoco es tan influyente como para que la claridad sea considerada como la mejor variable predictora para el precio. Por lo tanto, la relación no parece que sea significativa.

Las 2 últimas variables, en definitiva, tienen una gran variabilidad dentro de cada boxplot (nivel) y poca entre ellas.

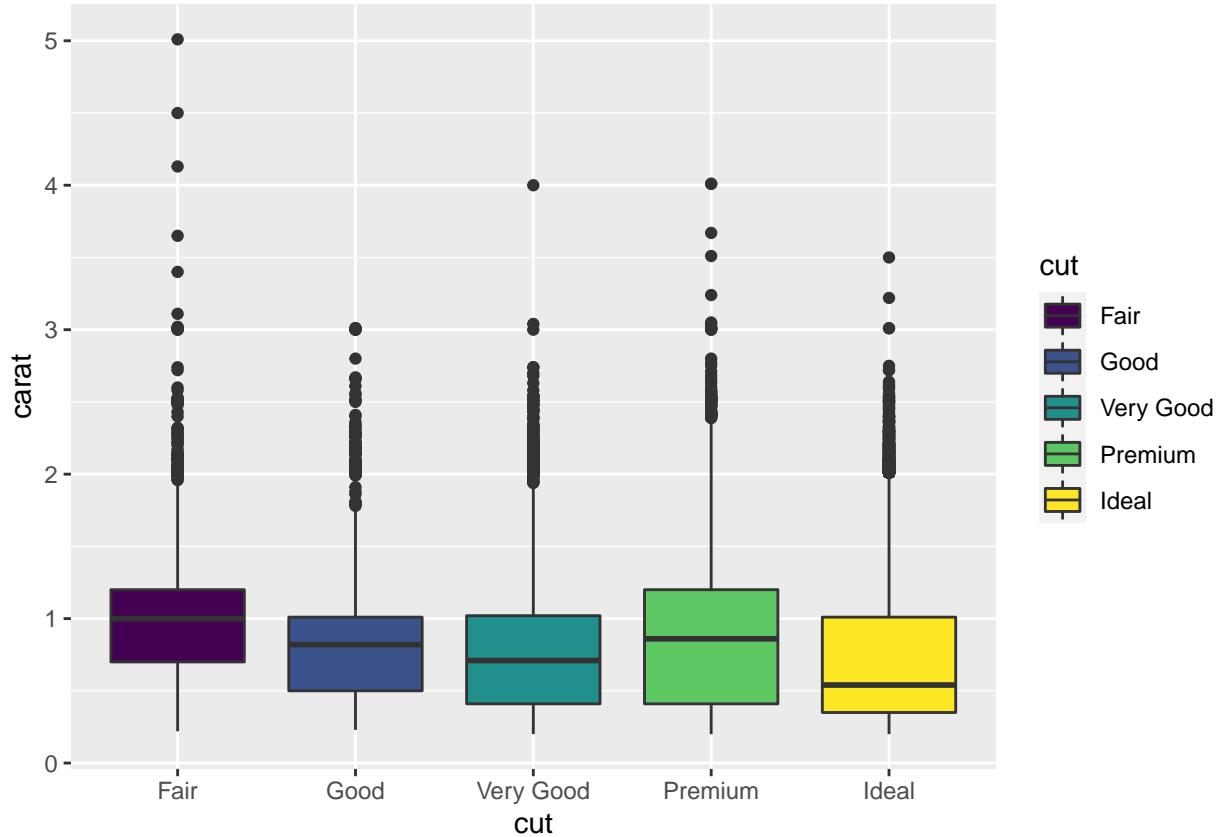
CONCLUSION

Teniendo en cuenta las consideraciones realizadas en cada uno de las representaciones gráficas, carat parece que es la mejor variable predictora de cara a conocer el precio de un diamante.

2) Como se relaciona la variable cut con el carat ?

Analizamos la correlación entre ambas variables a través de un boxplot, como hemos hecho hasta ahora en el caso de una variable continua con una categórica.

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



En este gráfico, se puede apreciar gran variabilidad en cuanto a los quilates en cada una de las categorías de corte, y en términos generales, una relación negativa leve entre ambas (variables), de forma que aquellos de más quilates, tienden a tener un corte de menor calidad (Fair).

3) ¿Por qué la combinación de estas dos relaciones hace que los diamantes de menor calidad sean más caros?

Lo que se puede concluir en este aspecto, con todo el conjunto de gráficos y representaciones elaboradas, es que cuanto menor es la calidad del corte, por lo general, la variable carat alcanza valores más elevados, influyendo de forma directa sobre el precio establecido para cada diamante, y provocando que aquellos de menor calidad acaben siendo más caros. Al final, en la primera pregunta, hemos respondido haciendo referencia al hecho de que los quilates son la mejor variable predictora de cara a conocer el precio del diamante.

- Haz el ejercicio 4 de la Sección 12.6.1 de R4DS. ¡Aprovecha el código previo de esa sección para trabajar con datos limpios!

Empezamos copiando el código previo al ejercicio para poder trabajar con datos limpios:

```
(who_Ejer=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))

## # A tibble: 76,046 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
## 7 Afghanistan  1997 sp     m    65     0
## 8 Afghanistan  1997 sp     f    014     5
## 9 Afghanistan  1997 sp     f   1524    38
## 10 Afghanistan 1997 sp    f   2534    36
## # ... with 76,036 more rows

```

Calculamos el numero total de casos, agrupados por sexo, país y año, obteniendo como resultado el siguiente DataFrame:

```

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

```

```

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

## # 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

```

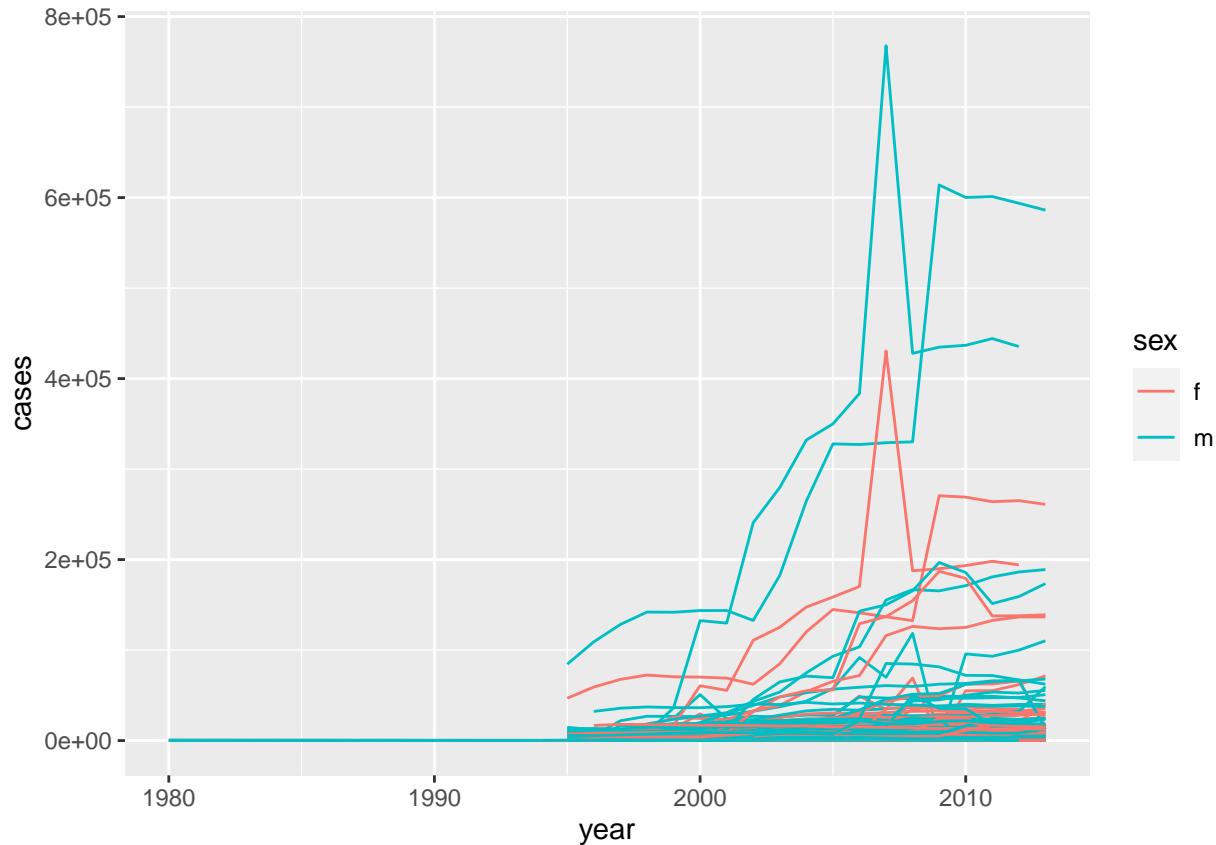
Para la visualización de la información que nos aporta la tabla realizada, haremos uso de un gráfico temporal como el que se verá a continuación.

Las columnas sex y country se juntan para que cada línea del gráfico represente la evolución temporal de los casos de TB en cada país para cada género.

```

who_TABLE %>%
  unite(pais_genero, country, sex, remove = FALSE) %>%
  ggplot(aes(x = year, y = cases, group = pais_genero, colour = sex)) +
  geom_line()

```



Claramente, se puede ver como, aproximadamente hasta el año 1995, hay países que no disponen de registros en el dataset o que el número de casos es muy reducido y constante, por lo que no interesa incluirlos en el estudio. Nos centraremos en las evoluciones que han sufrido a partir del 95.

```

(who_TABLE1=who_Ejer %>%
  group_by(country, year, sex) %>%
  filter(year > 1995) %>%
  summarise(cases = sum(cases)))

```

```

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

## # A tibble: 6,594 x 4
## # Groups:   country, year [3,320]
##   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 6,584 more rows

```

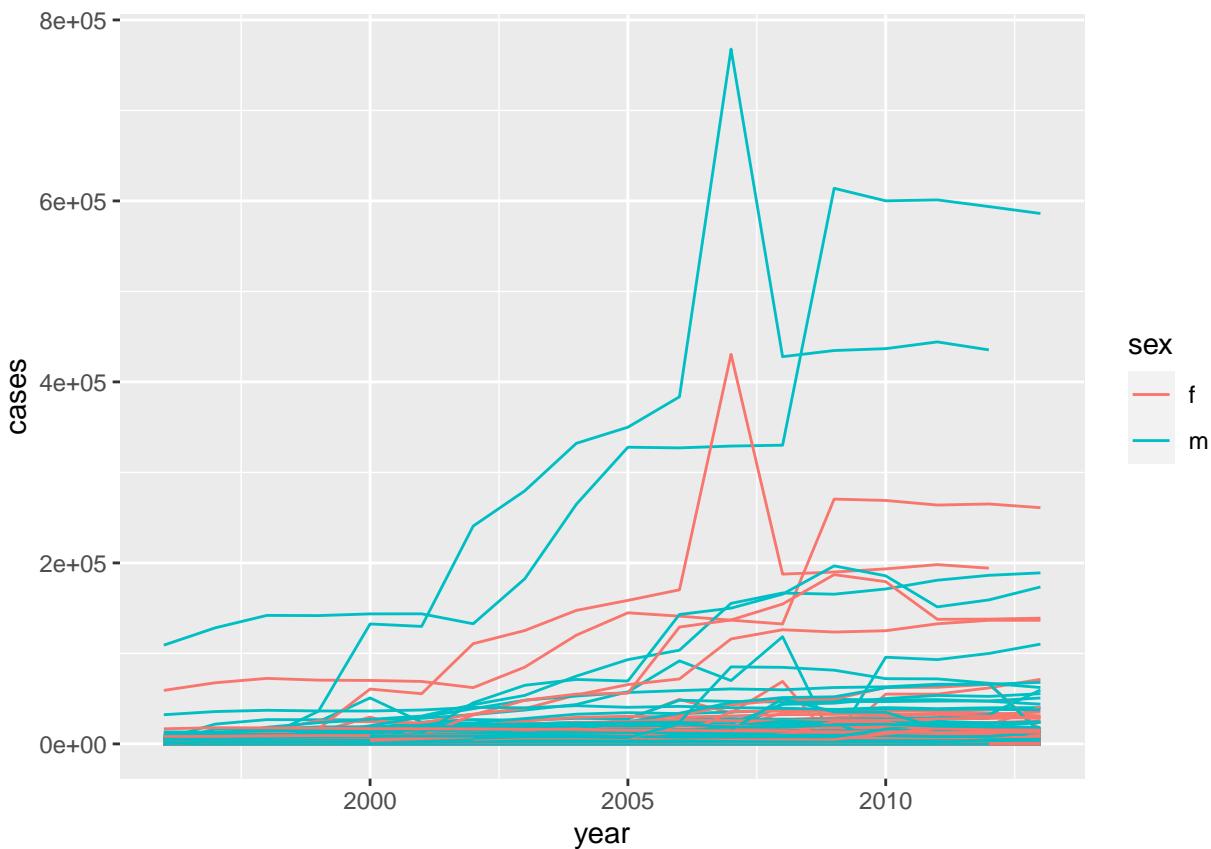
Trabajaremos con este nuevo DataFrame

Volvemos a realizar el gráfico temporal, pero centrándonos en el número de casos a partir del año 1995, como se ha comentado previamente.

```

who_TABLE1 %>%
  unite(pais_genero, country, sex, remove = FALSE) %>%
  ggplot(aes(x = year, y = cases, group = pais_genero, colour = sex)) +
  geom_line()

```



En este gráfico, vemos la evolución de cada sexo en cada país en lo referente a los casos de TB, aunque no se aprecia una distinción clara entre los países que están siendo representados.

Por lo tanto, a continuación se procede con la selección de aquellos que en total superan los 500K casos.

```

(paisesTopCases = who_Ejer %>%
  group_by(country) %>%
  summarise(n=sum(cases)) %>%
  filter(n>500000) %>%
  select(country))

```

```

## # A tibble: 16 x 1
##   country
##   <chr>
## 1 Bangladesh
## 2 Brazil
## 3 China
## 4 Democratic People's Republic of Korea
## 5 Democratic Republic of the Congo
## 6 Ethiopia
## 7 India
## 8 Indonesia
## 9 Kenya
## 10 Nigeria
## 11 Pakistan
## 12 Philippines
## 13 Russian Federation
## 14 South Africa
## 15 United Republic of Tanzania
## 16 Viet Nam

```

Finalmente, se representan los gráficos temporales para cada país (de los seleccionados previamente) en función del sexo, y cada uno con su escala correspondiente, para poder estudiar con claridad y alto nivel de detalle la evolución y tendencia de forma independiente.

```

who_TABLE1 %>%
  filter(country %in% paisesTopCases$country) %>%
  unite(pais_genero, country, sex, remove = FALSE) %>%
  ggplot(aes(x = year, y = cases, group = pais_genero, colour = sex)) +
  geom_line() +
  facet_wrap(~ country, scales='free_y')

```

