

Tarea 2. FMAD 2021-2022

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

Rodríguez González, Álvaro

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

Contents

| | |
|---|-----------|
| Ejercicio 1. Simulando variables aleatorias discretas. | 3 |
| Apartado 1 | 3 |
| Apartado 2 | 3 |
| Apartado 3 | 7 |
| Apartado 4 | 10 |
| Ejercicio 2. Datos limpios | 11 |
| Ejercicio 3. Lectura de R4DS. | 13 |
| Apartado 1 | 13 |
| Apartado 2 | 19 |

Ejercicio 1. Simulando variables aleatorias discretas.

Apartado 1

Enunciado: 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.

Solución: Lo primero de todo será saber como es la función para calcular la media teórica. Esta viene dada por la siguiente expresión:

$$\mathbb{E}(X) = \mu = \sum_{i=1}^n x_i * \mathbb{P}(x_i)$$

Por tanto, usando la expresión anterior tenemos:

```
(media_poblacional <- sum(c(0:3)*c(64/125,48/125,12/125,1/125)))
```

```
## [1] 0.6
```

En cuanto a la varianza esta tiene la siguiente expresión, que desarrollandola obtenemos una expresión más sencilla, que se usará para el cálculo.

$$\text{Var}(X) = \mathbb{E}[(X - \mu)^2] \quad \longrightarrow \quad \text{Var}(X) = \mathbb{E}(X^2) - \mathbb{E}(X)$$

```
(varianza_poblacional <- sum(c(0:3)^2*c(64/125,48/125,12/125,1/125)) - media_poblacional)
```

```
## [1] 0.24
```

Apartado 2

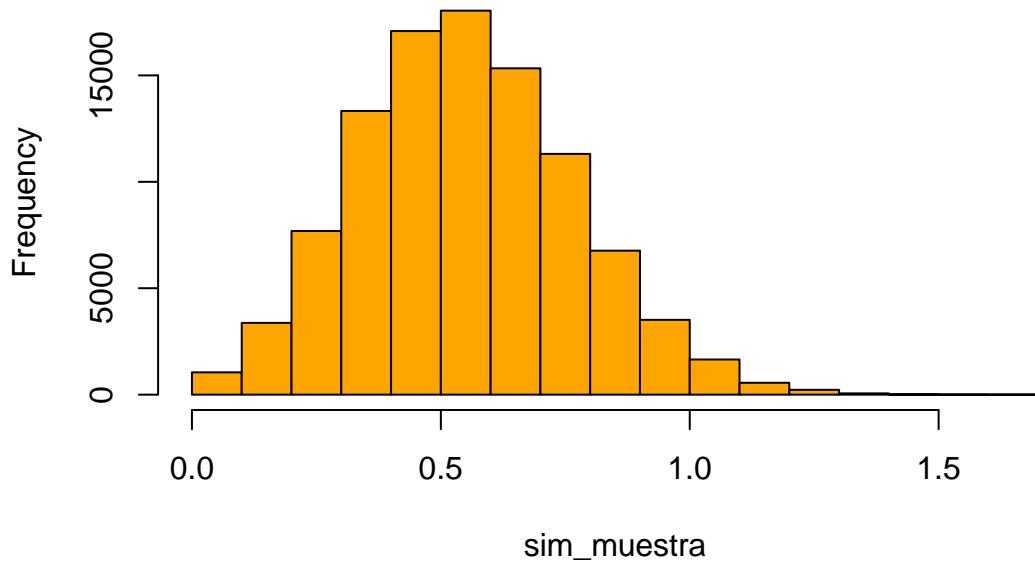
Enunciado: Combina `sample` con `replicate` para simular cien mil muestras de tamaño 10 de esta variable X_1 . Estudia la distribución de las medias muestrales como hemos hecho en ejemplos previos, ilustrando con gráficas la distribución de esas medias muestrales. Cambia después el tamaño de la muestra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

Solución: Comenzamos simulando cien mil muestras de tamaño 10 de la variable X_1 .

```
sim_muestra <- replicate(100000, {
  mean(sample(0:3,10,replace = TRUE, prob = c(64,48,12,1)))
})
```

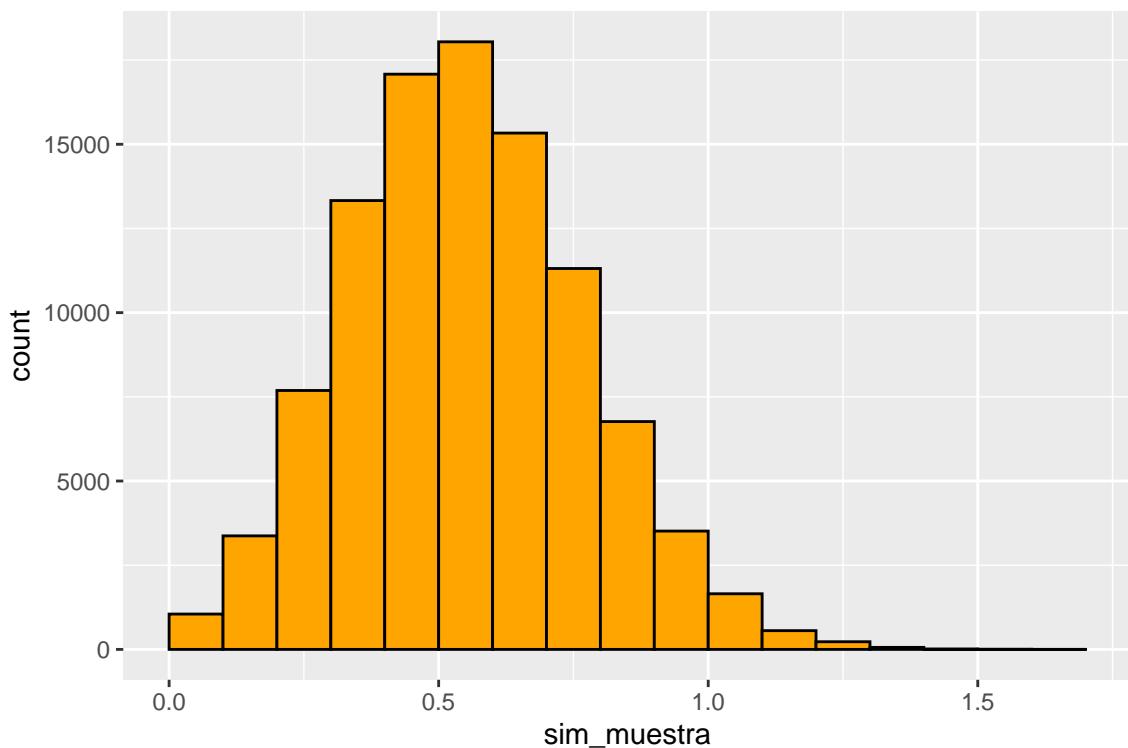
Ahora representamos las medias obtenidas de esta simulación a través de un histograma:

```
hist(sim_muestra, main = NULL ,col = "orange",)
```



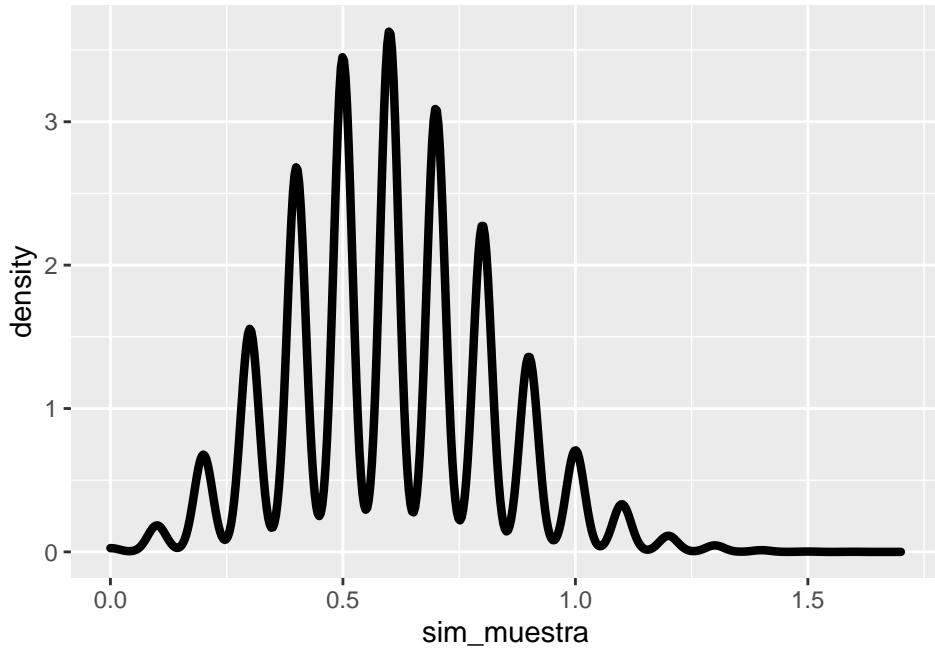
Otra forma de hacer el histograma es usando ggplot:

```
sim_muestra_dataframe <- data.frame(sim_muestra)
ggplot(sim_muestra_dataframe, mapping = aes(x = sim_muestra)) +
  geom_histogram(breaks = seq(min(sim_muestra_dataframe$sim_muestra),
    max(sim_muestra_dataframe$sim_muestra), length.out = 18),
    fill = "orange", color="black")
```



Después de hacer los histogramas podemos realizar una curva de densidad:

```
ggplot(sim_muestra_dataframe) +
  geom_density(mapping = aes(x = sim_muestra), color="black", size=1.5)
```

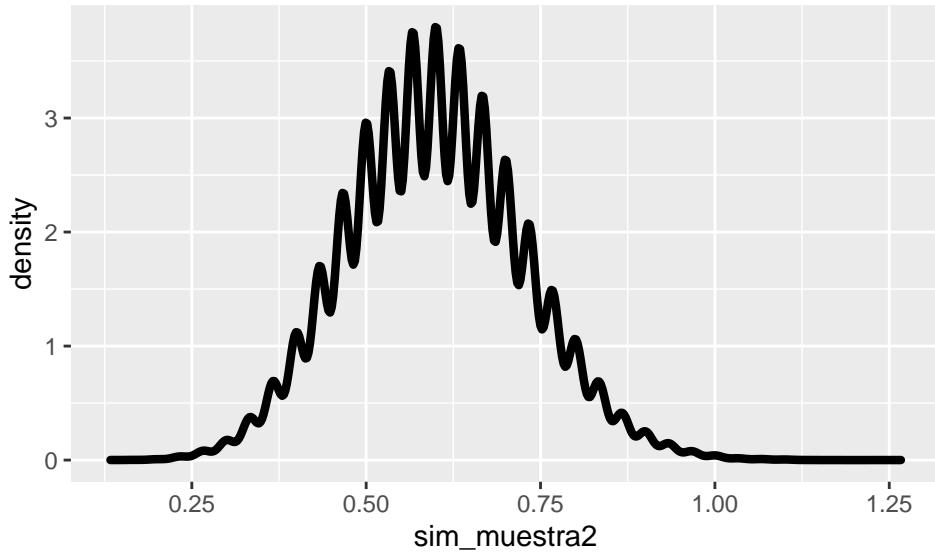


Ahora cambio el tamaño de la muestra a 30 y repito el mismo análisis hecho anteriormente.

```
sim_muestra2 <- replicate(100000, {
  mean(sample(0:3,30,replace = TRUE, prob = c(64,48,12,1)))
})
```

Y su vez repetimos el proceso que hemos hecho antes para ilustrar la muestra.

```
ggplot(sim_muestra2_dataframe) +
  geom_density(mapping = aes(x = sim_muestra2), color="black", size=1.5)
```



```

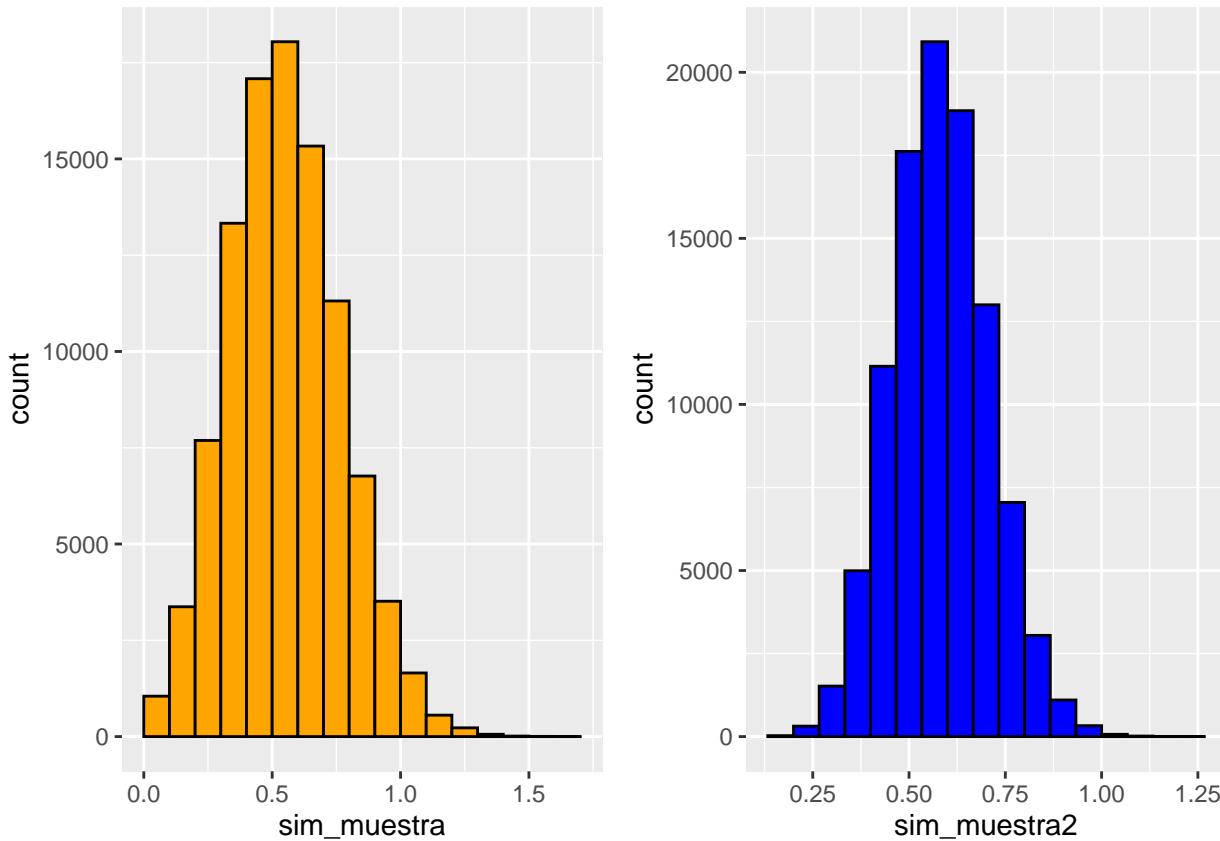
p1 <- ggplot(sim_muestra_dataframe, mapping = aes(x = sim_muestra)) +
  geom_histogram(breaks = seq(min(sim_muestra_dataframe$sim_muestra),
    max(sim_muestra_dataframe$sim_muestra), length.out = 18),
    fill = "orange", color="black")

sim_muestra2_dataframe <- data.frame(sim_muestra2)

p2 <- ggplot(sim_muestra2_dataframe, mapping = aes(x = sim_muestra2)) +
  geom_histogram(breaks = seq(min(sim_muestra2_dataframe$sim_muestra2),
    max(sim_muestra2_dataframe$sim_muestra2), length.out = 18),
    fill = "blue", color="black")

grid.arrange(p1,p2, nrow = 1)

```



Por último representamos nuestras curvas de densidad de ambas muestras.

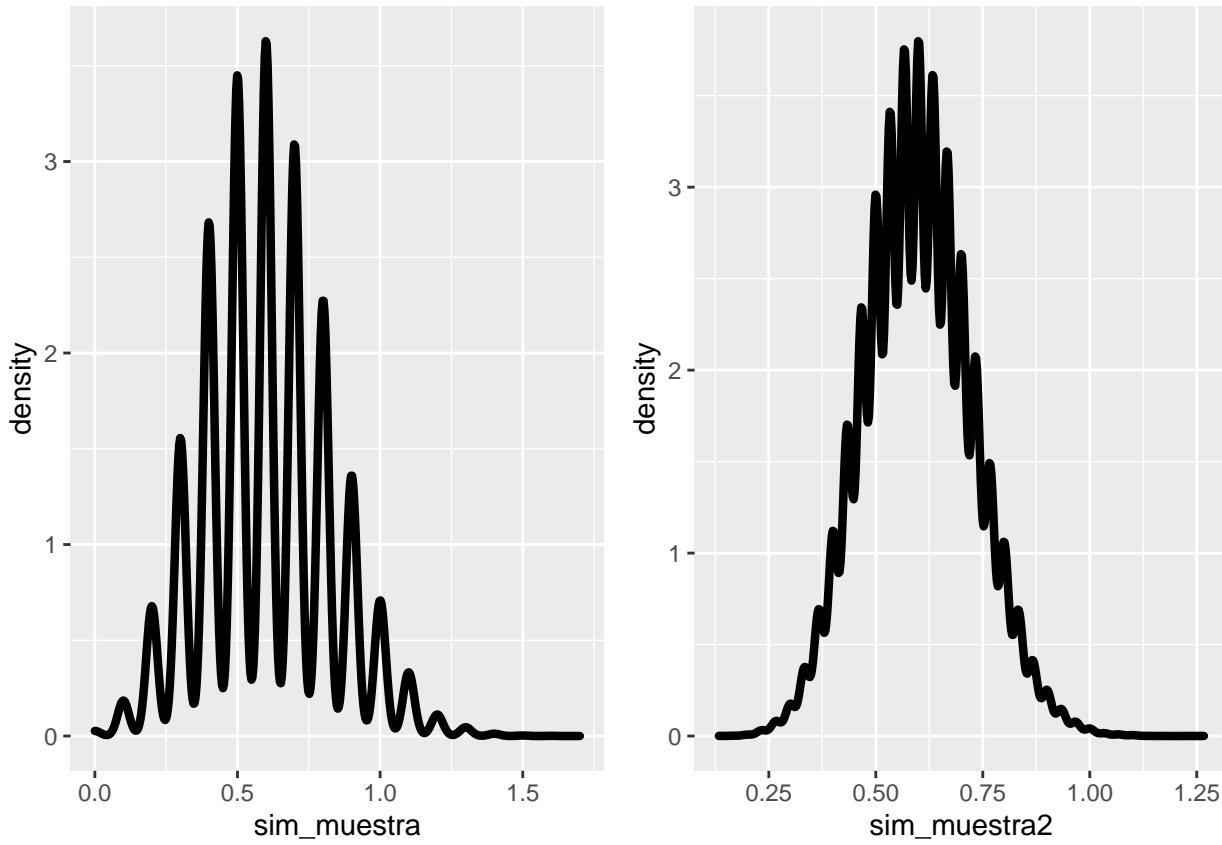
```

p11 <- ggplot(sim_muestra_dataframe) +
  geom_density(mapping = aes(x = sim_muestra), color="black", size=1.5)

p22 <- ggplot(sim_muestra2_dataframe) +
  geom_density(mapping = aes(x = sim_muestra2), color="black", size=1.5)

grid.arrange(p11,p22, nrow = 1)

```



Apartado 3

Enunciado: 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?

Solución: Vamos a responder por partes:

- ¿Qué valores puede tomar la suma $X_1 + X_2$?

La suma de la variable $X_1 + X_2$ puede tomar valores desde 0 hasta 5. Esto se debe a que la variable X_1 toma valores comprendidos entre 0 y 3 (ambos incluidos) y la variable X_2 toma valores comprendidos entre 0 y 2 (ambos incluidos).

- ¿Cuál es su tabla de probabilidad?

Para una notación más sencilla y resumida renombraremos a la variable $X_1 + X_2$ como Y . Por ello la tabla de probabilidad se nos queda de la siguiente forma (sin calcular ningún elemento de la tabla).

| | | | | | | |
|--|---|---|---|---|---|---|
| valor de Y | 0 | 1 | 2 | 3 | 4 | 5 |
| Probabilidad de ese valor $P(Y = y_i)$ | ? | ? | ? | ? | ? | ? |

Ahora comenzaremos a calcular las probabilidades que tenemos en la tabla:

- $\mathbb{P}(Y = 0)$ o $\mathbb{P}(X_1 + X_2 = 0)$

En este caso la única posibilidad es que tanto X_1 como X_2 sean 0. Esto se puede representar en lenguaje matematico como:

$$\mathbb{P}(X_1 = 0 \text{ y } X_2 = 0) \longrightarrow \mathbb{P}((X_1 = 0) \cap (X_2 = 0))$$

Ahora como X_1 y X_2 son variables independientes podemos afirmar lo siguiente:

$$\mathbb{P}((X_1 = 0) \cap (X_2 = 0)) = \mathbb{P}(X_1 = 0) \cdot \mathbb{P}(X_2 = 0)$$

Por ello, solo nos queda mirar a la tabla de ambas variables y calcular la expresión anterior. Pero antes vamos a crear dos vectores con las probabilidades de ambas variables:

```
prob1 = c(64/125,48/125,12/125,1/125)
prob2 <- c(1/2,1/4,1/4)
```

Calculamos $\mathbb{P}(Y = 0)$:

```
(py0 <- fractions(prob_0 <- prob1[1]*prob2[1]))
```

```
## [1] 32/125
```

- $\mathbb{P}(Y = 1)$ o $\mathbb{P}(X_1 + X_2 = 1)$

En este caso hay dos posibilidades, o que X_1 sea 1 y X_2 sea 0 o que X_1 sea 0 y X_2 sea 1. Calculamos ambas probabilidades:

```
(p10 <- fractions(prob1[2]*prob2[1]))
```

```
## [1] 24/125
```

```
(p01 <- fractions(prob1[1]*prob2[2]))
```

```
## [1] 16/125
```

La probabilidad de que la suma sea 1 es la suma de ambas probabilidades.

```
(py1 <- fractions(sum(c(p10,p01))))
```

```
## [1] 8/25
```

- $\mathbb{P}(Y = 2)$ o $\mathbb{P}(X_1 + X_2 = 2)$

Aquí volvemos a realizar el mismo procedimiento que antes, veamos que casos nos dan lo que buscamos. En primer lugar $X_1 = 2$ y $X_2 = 0$, $X_1 = 1$ y $X_2 = 1$ y $X_1 = 0$ y $X_2 = 2$. Aplicando independencia es muy sencillo sacar las probabilidades de cada uno:

```
(p20 <- fractions(prob1[3]*prob2[1]))  
## [1] 6/125  
(p11 <- fractions(prob1[2]*prob2[2]))  
## [1] 12/125  
(p02 <- fractions(prob1[1]*prob2[3]))  
## [1] 16/125
```

Volviendo a sumar las probabilidades obtenemos la probabilidad buscada.

```
(py2 <- fractions(sum(c(p20,p11,p02))))  
## [1] 34/125
```

- $\mathbb{P}(Y = 3)$ o $\mathbb{P}(X_1 + X_2 = 3)$

Como en los casos anteriores tenemos varias formas de que la suma de las variables sea 3. Las posibilidades son las siguientes: $X_1 = 3$ y $X_2 = 0$, $X_1 = 2$ y $X_2 = 1$ y $X_1 = 1$ y $X_2 = 2$.

Calculamos la probabilidad de cada uno de los casos:

```
(p30 <- fractions(prob1[4]*prob2[1]))  
## [1] 1/250  
(p21 <- fractions(prob1[3]*prob2[2]))  
## [1] 3/125  
(p12 <- fractions(prob1[2]*prob2[3]))  
## [1] 12/125
```

Y sumando todas:

```
(py3 <- fractions(sum(c(p30,p21,p12))))  
## [1] 31/250
```

- $\mathbb{P}(Y = 4)$ o $\mathbb{P}(X_1 + X_2 = 4)$

En este caso las posibilidades se van reduciendo ya que X_2 toma como máximo valor el 2. Repetimos el mismo procedimiento que en los casos anteriores.

```
(p31 <- fractions(prob1[4]*prob2[2]))  
## [1] 1/500  
(p22 <- fractions(prob1[3]*prob2[3]))  
## [1] 3/125
```

```
(py4 <- fractions(sum(c(p31,p22))))
```

```
## [1] 13/500
```

- $\mathbb{P}(Y = 5)$ o $\mathbb{P}(X_1 + X_2 = 5)$

Procedemos de forma análoga a las anteriores.

```
(py5 <- fractions(prob1[4]*prob2[3]))
```

```
## [1] 1/500
```

Ahora representamos en una tabla las probabilidades y posteriormente comprobamos que la suma de todas las probabilidades de 1.

| valor de Y | 0 | 1 | 2 | 3 | 4 | 5 |
|--|------------------|----------------|------------------|------------------|------------------|-----------------|
| Probabilidad de ese valor $P(Y = y_i)$ | $\frac{32}{125}$ | $\frac{8}{25}$ | $\frac{34}{125}$ | $\frac{31}{250}$ | $\frac{13}{500}$ | $\frac{1}{500}$ |

```
(sum(c(py0,py1,py2,py3,py4,py5))) == 1
```

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

Apartado 4

Enunciado: 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.

Solución: El cálculo de la media teórica de la suma $X_1 + X_2$ se hace de la misma forma que hicimos el apartado 1. Se debe usar la noción de esperanza matemática de la variable suma ($\mathbb{E}(X_1 + X_2)$):

```
proby <- c(py0,py1,py2,py3,py4,py5)
(sum(c(0:5)*proby))
```

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

Usaremos `sample` y `replicate` para simular cien mil valores de esta variable suma y calcularemos la media de dichos valores.

```
datos_repsam <- replicate(100000, {
  sample(0:5,1,replace = TRUE, prob = prob)})
```



```
mean(datos_repsam)
```

```
## [1] 1.35163
```

Ejercicio 2. Datos limpios

Enunciado: El fichero a descargar 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.

Solución:

Lo primero que haremos será descargar e importar el fichero de datos sobre el que trabajaremos:

```
testResults <- read_csv("data/testResults.csv")  
knitr::kable(head(testResults, 10))
```

| name | id | gender_age | test_number | week1 | week2 | week3 | week4 | week5 |
|-------------|-----|------------|-------------|-------|-------|-------|-------|-------|
| Jacob | 108 | m_20 | | 1 | 8 | 5 | 7 | 5 |
| Jacob | 108 | m_20 | | 2 | 2 | 2 | 4 | 0 |
| Michael | 490 | m_19 | | 1 | 10 | 0 | 5 | 4 |
| Michael | 490 | m_19 | | 2 | 9 | 10 | 8 | 10 |
| Matthew | 424 | m_18 | | 1 | 6 | 0 | 0 | 1 |
| Matthew | 424 | m_18 | | 2 | 3 | 4 | 2 | 5 |
| Joshua | 734 | m_17 | | 1 | 10 | 2 | 2 | 0 |
| Joshua | 734 | m_17 | | 2 | 10 | 0 | 6 | 8 |
| Christopher | 928 | m_20 | | 1 | 5 | 2 | 0 | 0 |
| Christopher | 928 | m_20 | | 2 | 9 | 9 | 3 | 10 |

Antes de comenzar vamos a recordar los principios del tidy data. Se dice que un conjunto de datos es limpio se cumple las siguientes tres condiciones:

- Cada variable tiene su propia columna.
- Cada observación tiene su propia fila.
- Cada valor tiene su propia celda.

Ahora vamos a ver porque esta tabla de datos no se considera limpia, iremos viendo una a una si se cumplen las condiciones enunciadas anteriormente:

- ¿Cada variable tiene su propia columna?

No, el claro ejemplo es que podemos encontrar que tanto la variable género como la variable edad están encuadradas en una misma columna. También tenemos el caso de la variable semanas, que aparece en la parte superior de forma expandida por semanas (lo mismo que ocurría en los apuntes con la variable meses).

Una vez visto donde se encontraban nuestros problemas vamos a solucionarlos para convertir a la tabla en una tabla de datos limpios. Para ello pasaremos las semanas a una columna que se llamará week. A su vez también separaremos la columna gender_age en dos columnas que serán la variable género (gender) y años (age).

```

prueba2 <- testResults %>%
  pivot_longer(week1:week5, names_to = "week") %>%
  separate(gender_age, c("gender", "age"), sep="_", convert = TRUE)
knitr::kable(head(prueba2, 10))

```

| name | id | gender | age | test_number | week | value |
|-------|-----|--------|-----|-------------|-------|-------|
| Jacob | 108 | m | 20 | 1 | week1 | 8 |
| Jacob | 108 | m | 20 | 1 | week2 | 5 |
| Jacob | 108 | m | 20 | 1 | week3 | 7 |
| Jacob | 108 | m | 20 | 1 | week4 | 5 |
| Jacob | 108 | m | 20 | 1 | week5 | 6 |
| Jacob | 108 | m | 20 | 2 | week1 | 2 |
| Jacob | 108 | m | 20 | 2 | week2 | 2 |
| Jacob | 108 | m | 20 | 2 | week3 | 4 |
| Jacob | 108 | m | 20 | 2 | week4 | 0 |
| Jacob | 108 | m | 20 | 2 | week5 | 3 |

Ejercicio 3. Lectura de R4DS.

Apartado 1

Enunciado: Haz el ejercicio 2 de la Sección 7.5.1.1 de R4DS. ¿Qué variable del conjunto de datos de diamantes es más importante para predecir el precio de un diamante? ¿Cómo se correlaciona esa variable con el corte? ¿Por qué la combinación de esas dos relaciones hace que los diamantes de menor calidad sean más caros?

Solución: Lo primero que vamos a hacer es ver la tabla y conocer con que variables trabajamos.

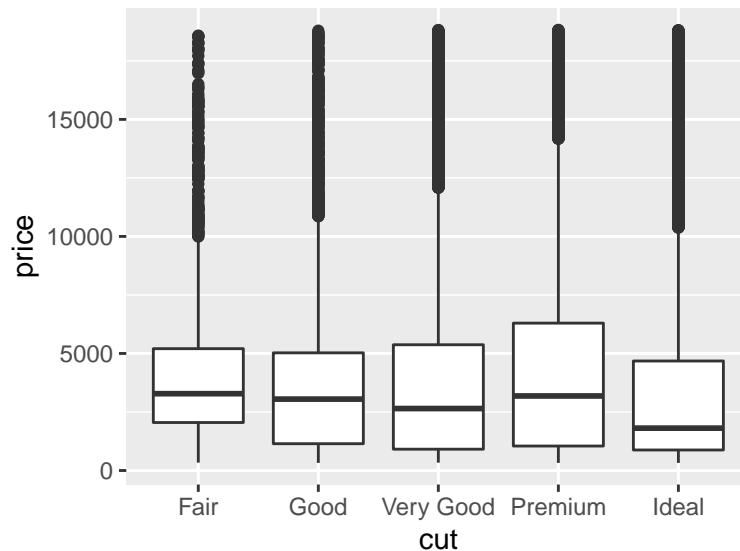
```
knitr::kable(head(diamonds, 6))
```

| carat | cut | color | clarity | depth | table | price | x | y | z |
|-------|-----------|-------|---------|-------|-------|-------|------|------|------|
| 0.23 | Ideal | E | SI2 | 61.5 | 55 | 326 | 3.95 | 3.98 | 2.43 |
| 0.21 | Premium | E | SI1 | 59.8 | 61 | 326 | 3.89 | 3.84 | 2.31 |
| 0.23 | Good | E | VS1 | 56.9 | 65 | 327 | 4.05 | 4.07 | 2.31 |
| 0.29 | Premium | I | VS2 | 62.4 | 58 | 334 | 4.20 | 4.23 | 2.63 |
| 0.31 | Good | J | SI2 | 63.3 | 58 | 335 | 4.34 | 4.35 | 2.75 |
| 0.24 | Very Good | J | VVS2 | 62.8 | 57 | 336 | 3.94 | 3.96 | 2.48 |

Vamos a estudiar como se comportan las diferentes variables cualitativas con la variable precio. Esto lo haremos con los histogramas donde agruparemos los datos en torno a la variable que estemos estudiando, como serán el corte, color y claridad.

- Corte y precio

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

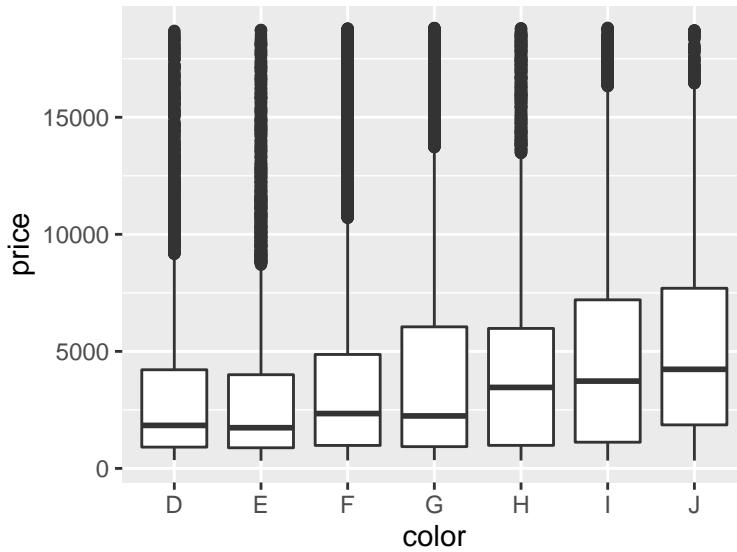


En un primer vistazo al histograma podemos afirmar que no existe correlación entre el tipo de corte de los diamantes y el precio. Esto se ve muy bien en que los diferentes tipos de corte presentan precios medios muy

similares (incluso valiendo más aquellos que peor cortados están). También no se puede afirmar que exista correlación entre ambas variables debido a la gran variabilidad de los datos.

- Color y precio

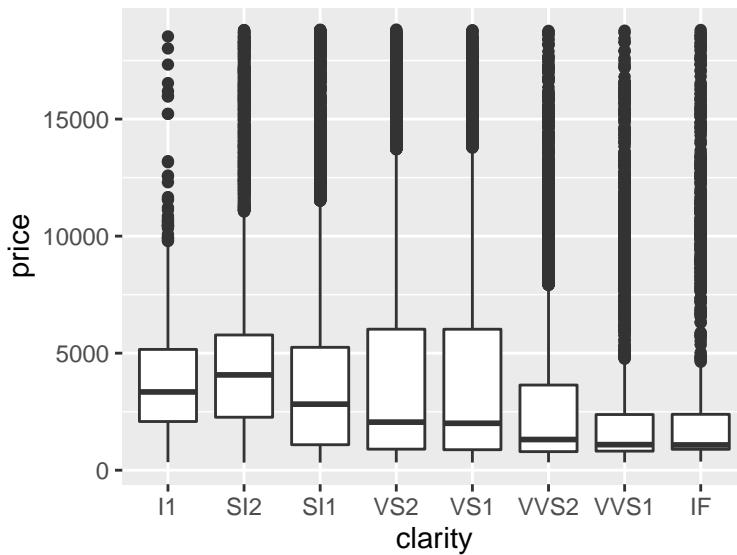
```
ggplot(diamonds, mapping = aes(x = color, y= price)) +
  geom_boxplot()
```



En este caso en un primer vistazo nos puede llamar la atención que cuanto más lejana es la letra el coste medio aumenta, lo que nos podría hacer pensar que existe una correlación positiva entre ambas variables. Pero esto no es así, debido principalmente a lo mismo que en el caso anterior. Existe una gran variabilidad de los datos por lo que no podemos afirmar nada de lo comentado anteriormente.

- Claridad y precio

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

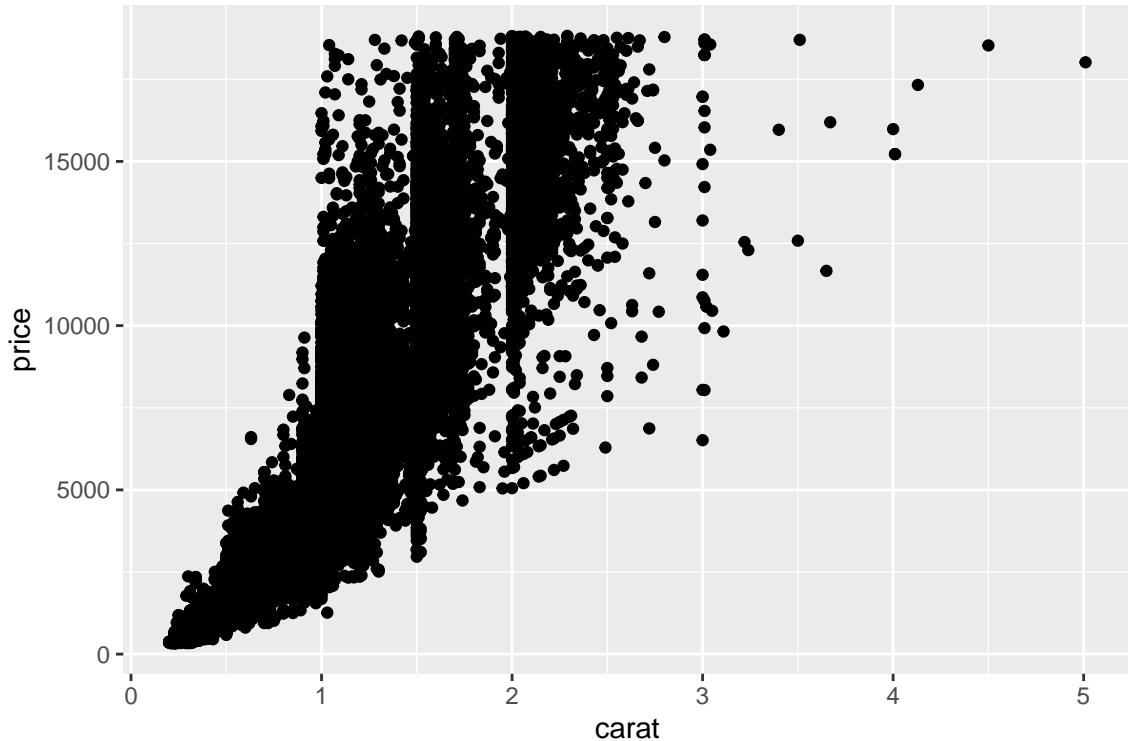


Esta última variable cualitativa presenta los mismos problemas que la variable corte, no se presenta ningún tipo relación con la variable precio debido a la gran variabilidad de los datos y una serie de medias muy similares.

Una vez visto y analizado como se comportan todas las variables cualitativas con la variable precio vamos a pasar a ver que ocurre con las variables cuantitativas como son los quilates, la profundidad y tres variables más.

- Quilates y precio

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



A diferencia de las variables vistas hasta ahora, la variable quilates presenta una fuerte correlación con la variable precio, ya que como podemos observar a mayor número de quilates el diamante tiene mayor precio. Además al tratarse de una variable cuantitativa podemos calcular la correlación de ambas variables.

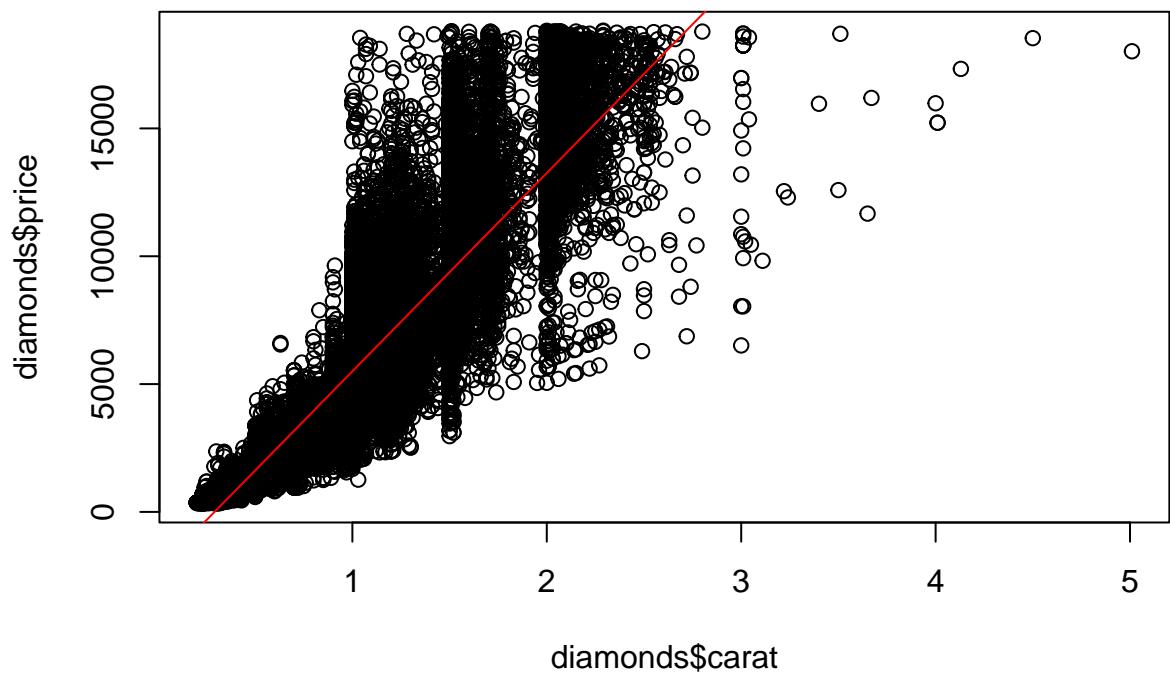
```
cor(diamonds$carat, diamonds$price)
```

```
## [1] 0.9215913
```

Una correlación de 0.92 indica un alto grado de ajuste lineal del precio en términos del número de quilates. Por tanto, con un aumento en el número de quilates esperamos un incremento en el precio.

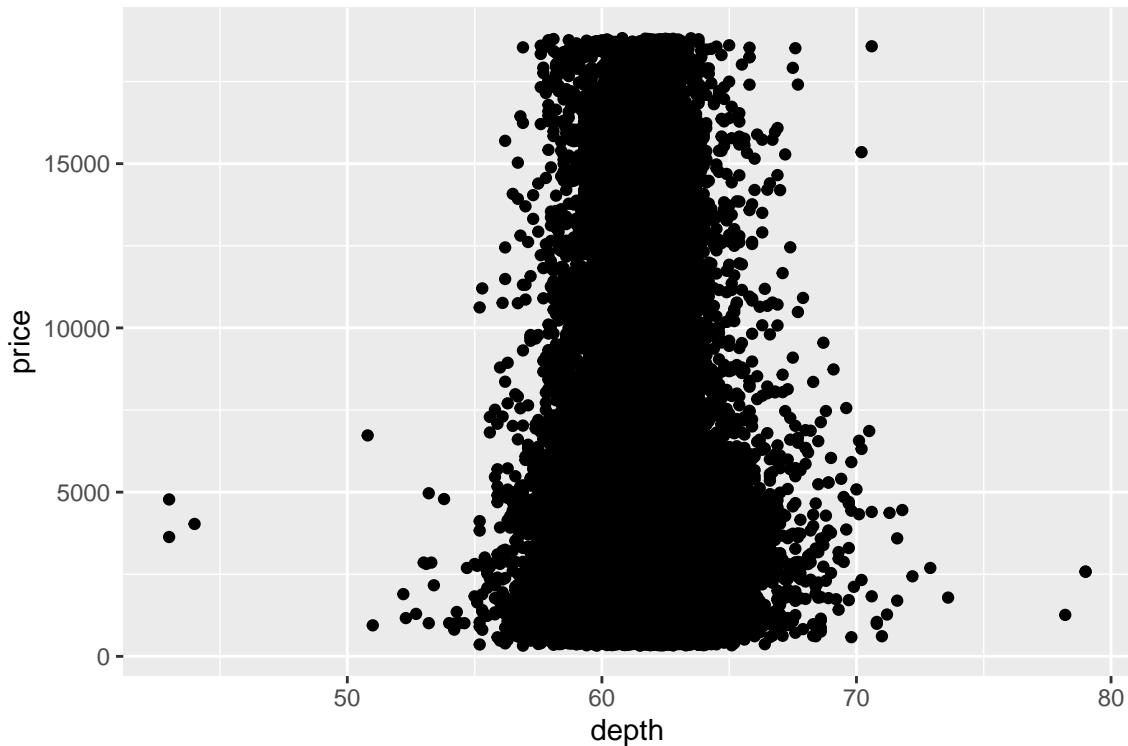
También si lo deseamos podemos ver la recta de regresión de como la variable quilate determina el precio.

```
coefs <- lm(diamonds$price ~ diamonds$carat)
plot(diamonds$carat, diamonds$price)
abline(coefs, col = "red")
```



- Profundidad y precio

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



De un simple vistazo podemos afirmar que la variable profundidad no tiene ningún tipo de correlación con la variable precio. Esto se debe a que los valores de la variable profundidad se encuentran entre 55 y 70, y lo que determina el precio no será la profundidad si no otra variable ya que no se observa que un aumento de la

profundidad aumente o disminuya el precio.

También podemos calcular el coeficiente de correlación lineal de ambas variables.

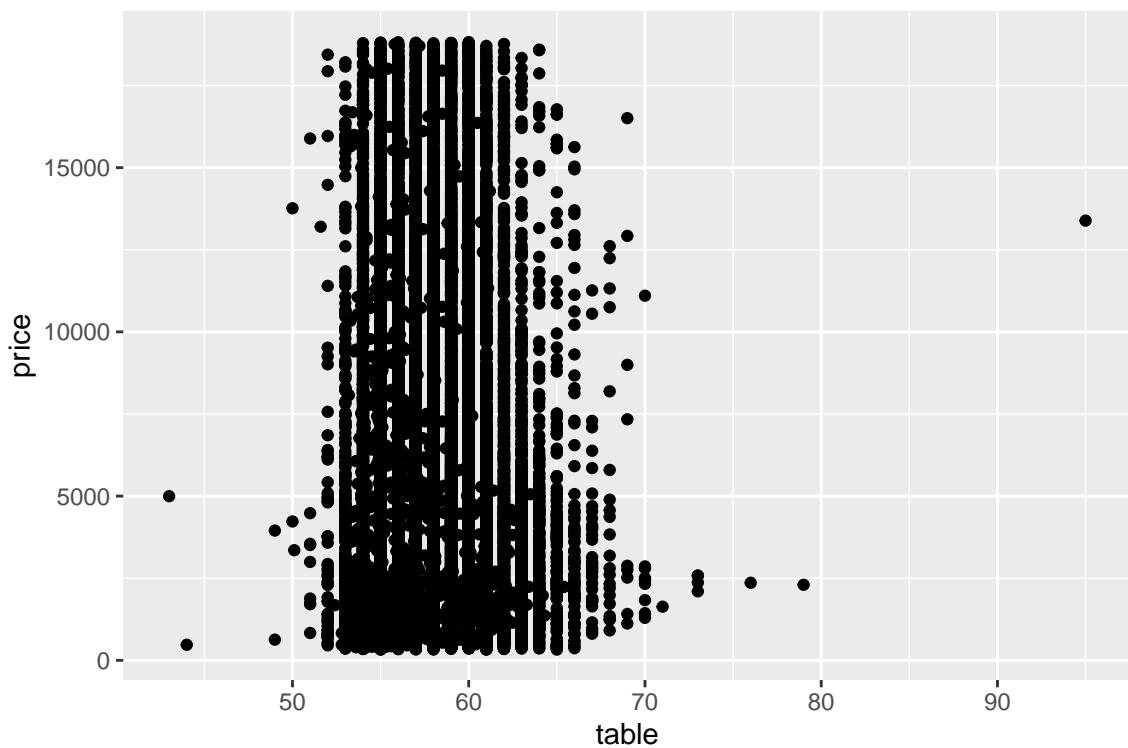
```
cor(diamonds$depth, diamonds$price)
```

```
## [1] -0.0106474
```

El coeficiente es cercano a 0, por lo que nos esta indicando que no existe ninguna relación entre ambas variables.

- Tabla y precio

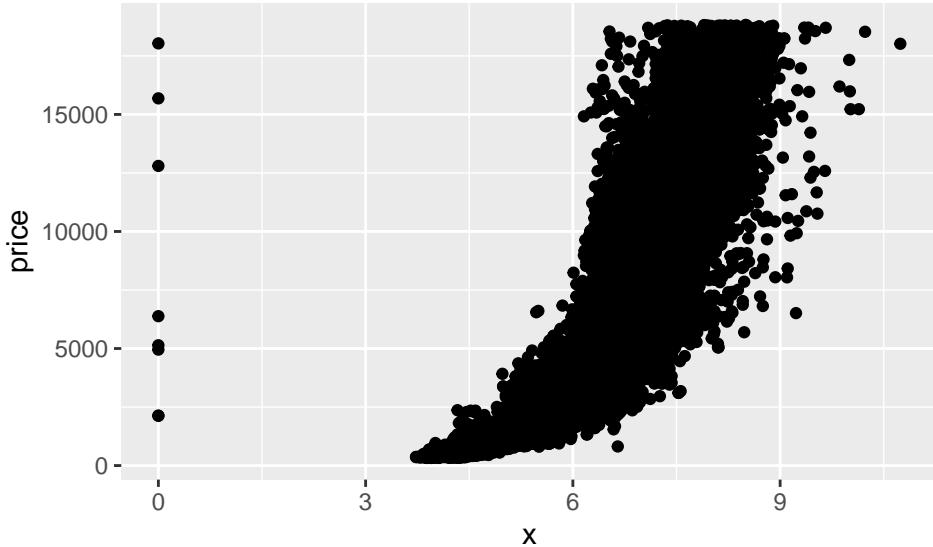
```
ggplot(diamonds, aes(x = table, y = price)) +  
  geom_point()
```



Este caso es similar al de la profundidad y el precio, por tanto podemos afirmar que no existe correlación entre estas variables.

- X y precio

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

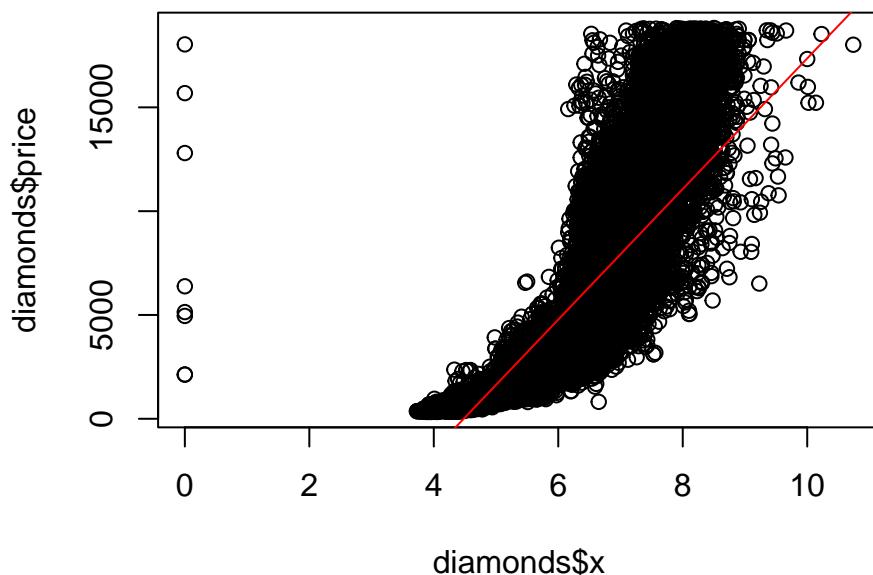


A simple vista podemos observar que un aumento en el valor de la variable x supone un aumento en el precio. Se puede interpretar que el crecimiento es de tipo exponencial debido a la curva que forman los puntos representados. Para confirmar que existe tal grado de correlación vamos a calcular tanto su coeficiente de correlación lineal y su recta de regresión.

```
(cor(diamonds$x, diamonds$price))

## [1] 0.8844352

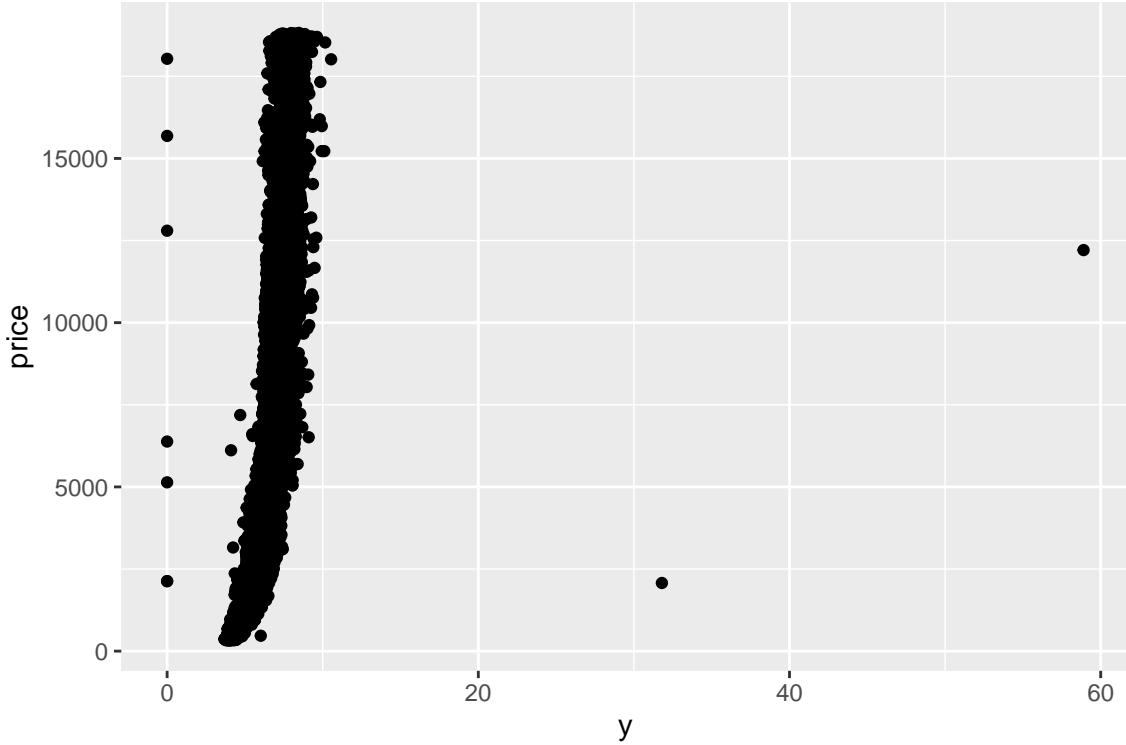
coefs_xp <- lm(diamonds$price ~ diamonds$x)
plot(diamonds$x, diamonds$price)
abline(coefs_xp, col = "red")
```



Tras estos dos cálculos podemos afirmar que existe un alto grado de ajuste lineal del precio en términos de la variable x. Pero si cabe destacar que a pesar de que esta variable influye en el precio tiene un mayor peso el número de quilates.

- Y y precio

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



Este caso es muy parecido al de la profundidad, por tanto no hace falta explicar nada.

Apartado 2

Enunciado: Haz el ejercicio 4 de la Sección 12.6.1 de R4DS. Para cada país, año y sexo, calcule el número total de casos de TB. Realiza una visualización informativa de los datos.

Solución: Lo primero será ver los datos de los que disponemos y si fuera necesario limpiarlos.

```
head(who, 10)
```

```
## # A tibble: 10 x 60
##   country     iso2   iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534 new_sp_m3544
##   <chr>      <chr>  <chr>  <int>      <int>      <int>      <int>      <int>
## 1 Afghanistan AF    AFG    1980       NA        NA        NA        NA
## 2 Afghanistan AF    AFG    1981       NA        NA        NA        NA
## 3 Afghanistan AF    AFG    1982       NA        NA        NA        NA
## 4 Afghanistan AF    AFG    1983       NA        NA        NA        NA
## 5 Afghanistan AF    AFG    1984       NA        NA        NA        NA
## 6 Afghanistan AF    AFG    1985       NA        NA        NA        NA
## 7 Afghanistan AF    AFG    1986       NA        NA        NA        NA
## 8 Afghanistan AF    AFG    1987       NA        NA        NA        NA
## 9 Afghanistan AF    AFG    1988       NA        NA        NA        NA
## 10 Afghanistan AF   AFG    1989       NA        NA        NA        NA
```

```

##  9 Afghanistan AF      AFG      1988          NA          NA          NA          NA
## 10 Afghanistan AF      AFG      1989          NA          NA          NA          NA
## # ... with 52 more variables: new_sp_m4554 <int>, new_sp_m5564 <int>,
## #   new_sp_m65 <int>, new_sp_f014 <int>, new_sp_f1524 <int>,
## #   new_sp_f2534 <int>, new_sp_f3544 <int>, new_sp_f4554 <int>,
## #   new_sp_f5564 <int>, new_sp_f65 <int>, new_sn_m014 <int>,
## #   new_sn_m1524 <int>, new_sn_m2534 <int>, new_sn_m3544 <int>,
## #   new_sn_m4554 <int>, new_sn_m5564 <int>, new_sn_m65 <int>,
## #   new_sn_f014 <int>, new_sn_f1524 <int>, new_sn_f2534 <int>, ...

```

Vemos que los datos no son datos limpios por lo que vamos a hacer las modificaciones necesarias para que podamos trabajar correctamente con ellos.

```

# Uso dplyr::select porque en la libreria MASS tambien existe una función select y si
# no especifico da error.
who1 <- who %>%
  pivot_longer(c(new_sp_m014:newrel_f65), names_to = "key", values_drop_na = TRUE) %>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel")) %>%
  separate(key, c("new", "type", "gender_age"), sep = "_") %>%
  separate(gender_age, c("gender", "age"), sep = 1) %>%
  dplyr::select(country, year, type, gender, age, value)
knitr::kable(head(who1, 7))

```

| country | year | type | gender | age | value |
|-------------|------|------|--------|------|-------|
| Afghanistan | 1997 | sp | m | 014 | 0 |
| Afghanistan | 1997 | sp | m | 1524 | 10 |
| Afghanistan | 1997 | sp | m | 2534 | 6 |
| Afghanistan | 1997 | sp | m | 3544 | 3 |
| Afghanistan | 1997 | sp | m | 4554 | 5 |
| Afghanistan | 1997 | sp | m | 5564 | 2 |
| Afghanistan | 1997 | sp | m | 65 | 0 |

Veamos como han ido evolucionando los diferentes tipos de detección de la enfermedad:

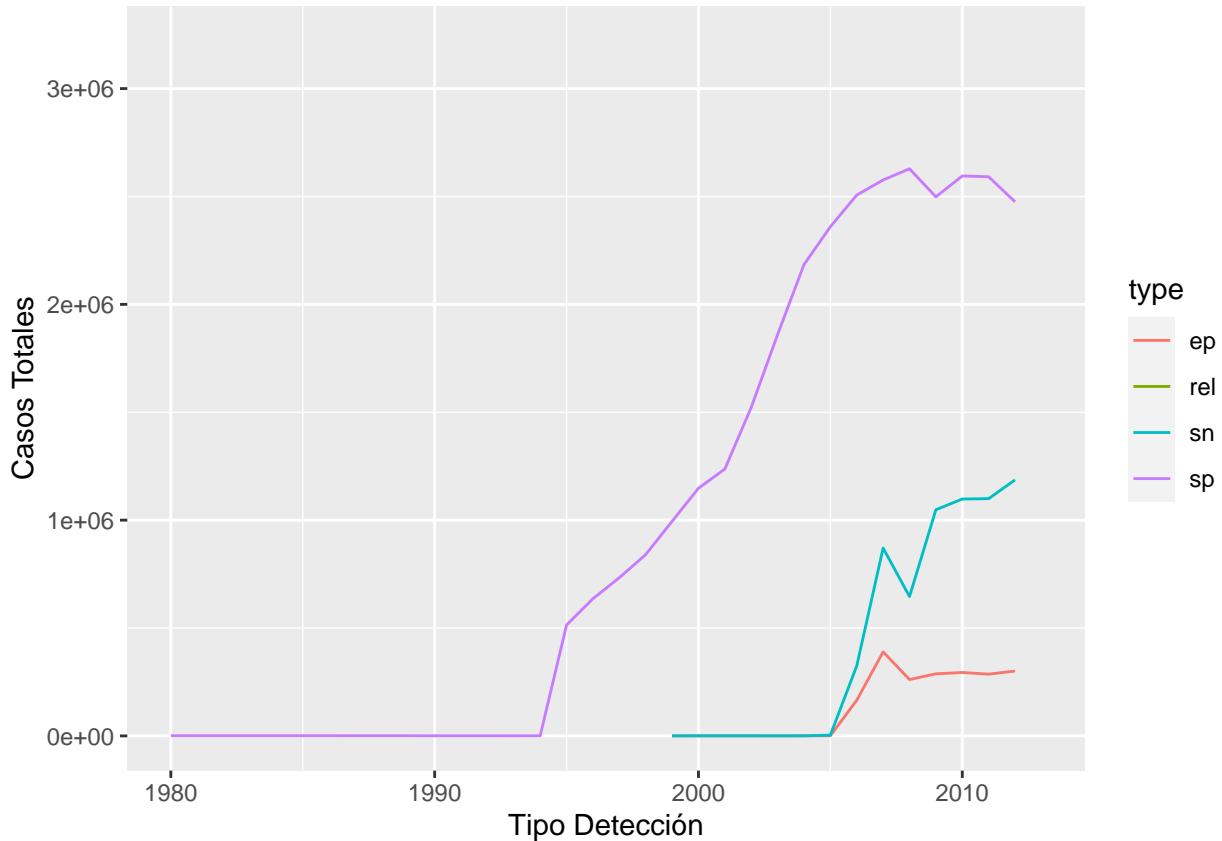
```

datos_tip_detecc <- who1 %>%
  dplyr::select(year, type, value) %>%
  group_by(year, type) %>%
  summarise(tot = sum(value))

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

ggplot(datos_tip_detecc, aes(x = year, y = tot)) +
  geom_freqpoly(stat = "identity", aes(color = type)) +
  xlab("Tipo Detección") +
  ylab("Casos Totales")

```



```

p1 <- ggplot(datos_pais_1, aes(x = year,y=tot)) +
  geom_freqpoly(stat = "identity", aes(color = gender)) +
  xlab("Año") + ylab("Casos") + ggtitle(lista_paises_top[1])

p2 <- ggplot(datos_pais_2, aes(x = year,y=tot)) +
  geom_freqpoly(stat = "identity", aes(color = gender)) +
  xlab("Año") + ylab("Casos") + ggtitle(lista_paises_top[2])

p3 <- ggplot(datos_pais_3, aes(x = year,y=tot)) +
  geom_freqpoly(stat = "identity", aes(color = gender)) +
  xlab("Año") + ylab("Casos") + ggtitle(lista_paises_top[3])

p4 <- ggplot(datos_pais_4, aes(x = year,y=tot)) +
  geom_freqpoly(stat = "identity", aes(color = gender)) +
  xlab("Año") + ylab("Casos") + ggtitle(lista_paises_top[4])

p5 <- ggplot(datos_pais_5, aes(x = year,y=tot)) +
  geom_freqpoly(stat = "identity", aes(color = gender)) +
  xlab("Año") + ylab("Casos") + ggtitle(lista_paises_top[5])

p6 <- ggplot(datos_pais_6, aes(x = year,y=tot)) +

```

```

geom_freqpoly(stat = "identity", aes(color = gender)) +
xlab("Año") + ylab("Casos") + ggtitle(lista_paises_top[6])

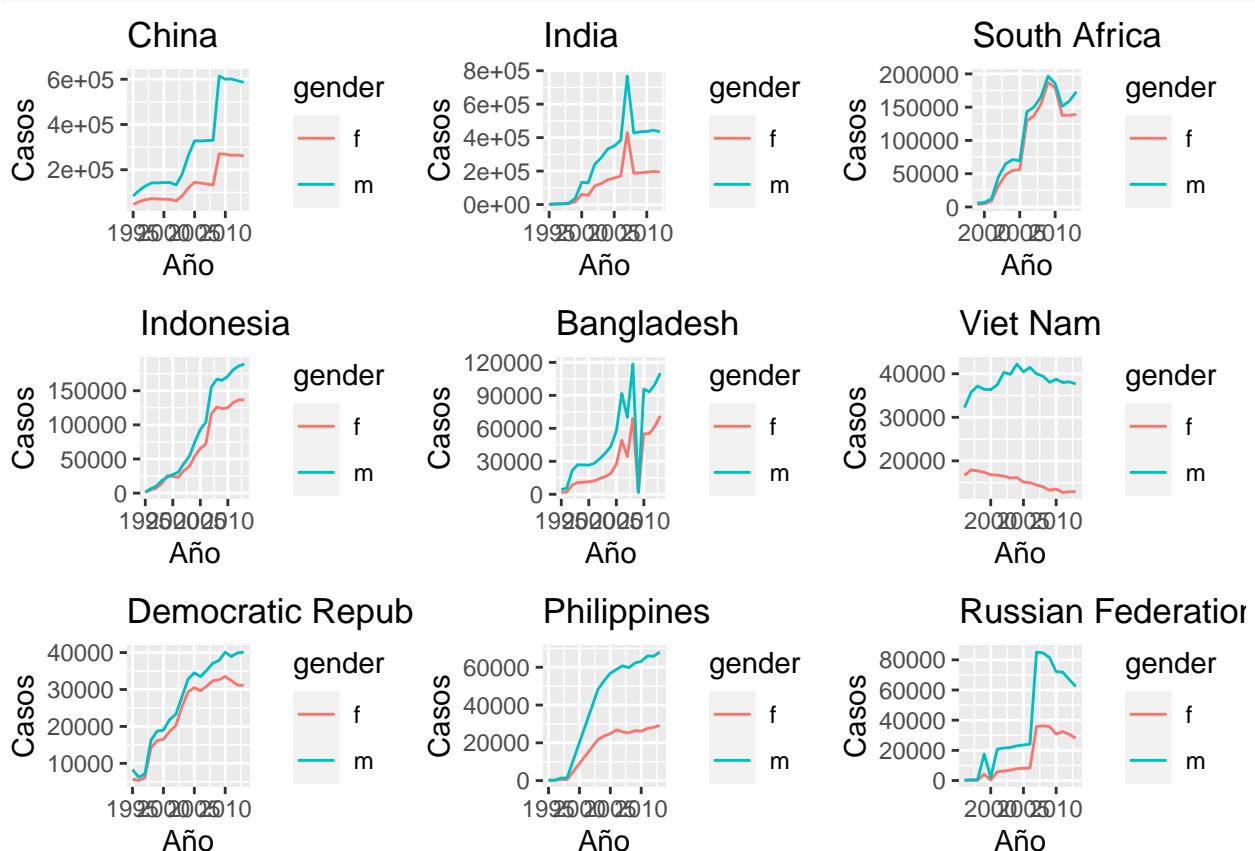
p7 <- ggplot(datos_pais_7, aes(x = year, y=tot)) +
geom_freqpoly(stat = "identity", aes(color = gender)) +
xlab("Año") + ylab("Casos") + ggtitle(lista_paises_top[7])

p8 <- ggplot(datos_pais_8, aes(x = year, y=tot)) +
geom_freqpoly(stat = "identity", aes(color = gender)) +
xlab("Año") + ylab("Casos") + ggtitle(lista_paises_top[8])

p9 <- ggplot(datos_pais_9, aes(x = year, y=tot)) +
geom_freqpoly(stat = "identity", aes(color = gender)) +
xlab("Año") + ylab("Casos") + ggtitle(lista_paises_top[9])

grid.arrange(p1,p2,p3,p4,p5,p6,p7,p8,p9, nrow = 3)

```



También puede resultar interesante ver como ha ido evolucionando la enfermedad en los grupos de edad a nivel mundial.

```

datos_evo_edad <- who1 %>%
dplyr::select(year, age, value) %>%

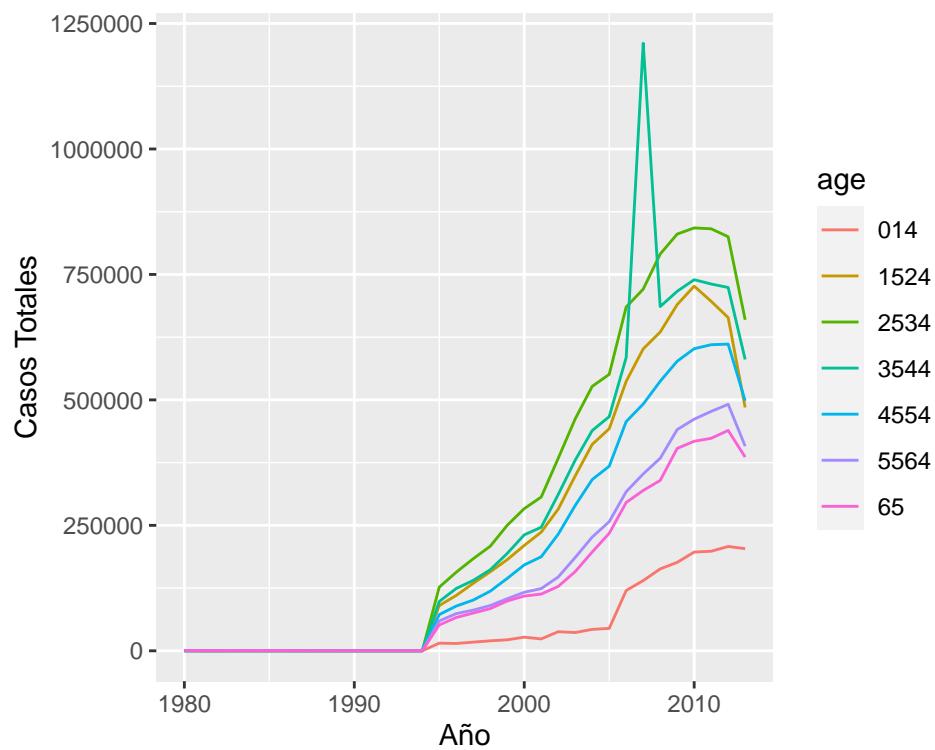
```

```

group_by(year, age) %>%
summarise(tot = sum(value))

ggplot(datos_evo_edad, aes(x = year, y = tot)) +
  geom_line(stat = "identity", aes(color = age)) +
  xlab("Año") +
  ylab("Casos Totales")

```



Para terminar vamos a trabajar con el primer país que aparece en la tabla, Afganistán. Sacaremos diferentes datos que pueden ser de utilidad.

El primero de ellos será la evolución de la enfermedad en el país a lo largo de los años.

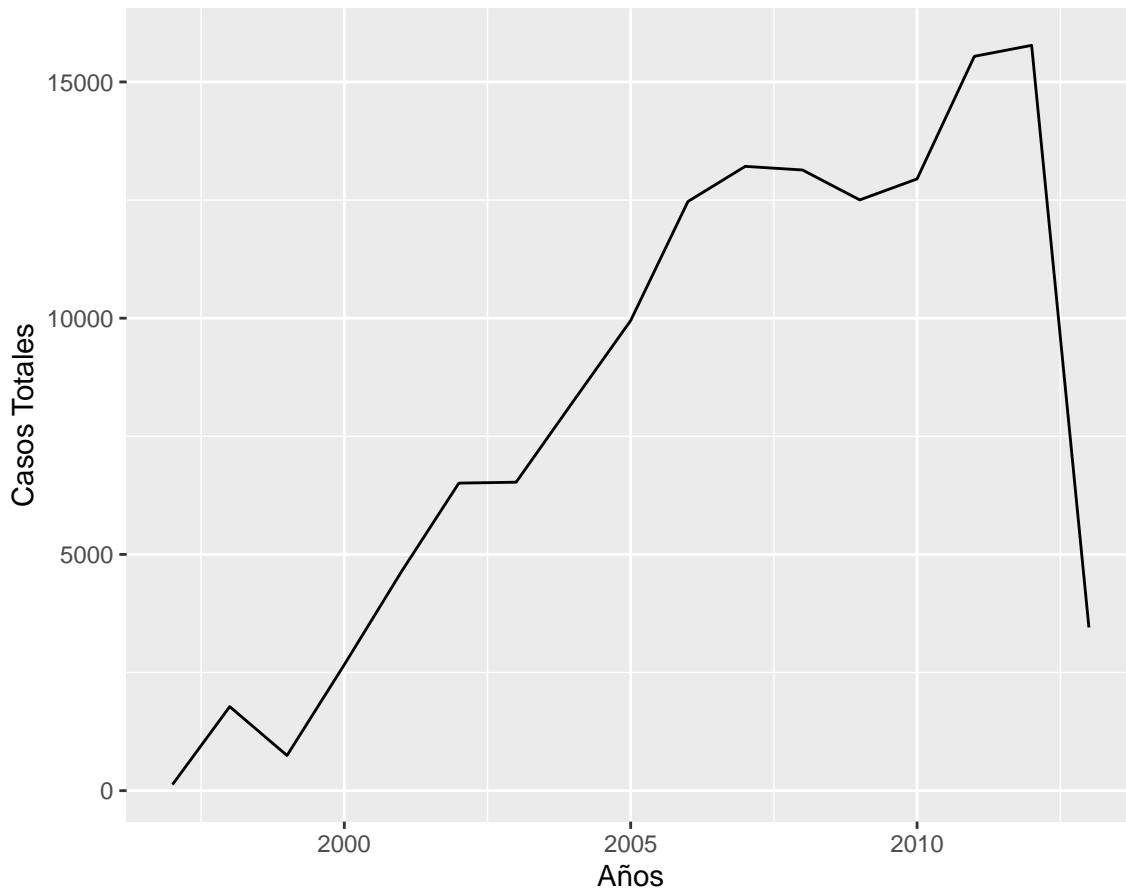
```

afga <- who1 %>%
  filter(country == "Afghanistan") %>%
  group_by(year) %>%
  summarise(Casos_Totales = sum(value))

ggplot(afga,aes(x = year, y = Casos_Totales)) +
  geom_freqpoly(stat = "identity") +
  xlab("Años") + ylab("Casos Totales") +
  ggtitle("Evolución Afganistán") +
  theme(plot.title = element_text(hjust = 0.5))

```

Evolución Afganistán



Ahora veremos como ha sido el comportamiento de la enfermedad a lo largo de los años distinguiendo por género.

```
afga_evo_genr <- who1 %>%
  filter(country == "Afghanistan") %>%
  dplyr::select(year, gender, value) %>%
  group_by(year, gender) %>%
  summarise(tot = sum(value))

ggplot(afga_evo_genr, aes(x = year, y = tot)) +
  geom_bar(stat = "identity", position=position_dodge(),
  aes(fill = gender), color = "black") +
  ylab("Casos Totales") +
  xlab("Año")
```

