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

Tarea 2

Gutiérrez García, Laura

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.

# Librerías

library(tidyverse)

# Ejercicio 1. Simulando variables aleatorias discretas.

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

$$
\begin{array}{|c|c|c|c|c|c|c|}
\hline
\text{valor de }X\_1 & 0 & 1 & 2 & 3 \\
\hline
\text{Probabilidad de ese valor }P(X = x\_i) & \dfrac{64}{125} &
\dfrac{48}{125}& \dfrac{12}{125} & \dfrac{1}{125}\rule{0mm}{6mm} \\[3mm]
\hline
\end{array}
$$

Calcula la media y la varianza teóricas de esta variable.

**Respuesta:**

La media teórica se calcula a partir de la tabla de probabilidades:

Y para la varianza teórica:

Almacenamos en dos vectores los valores correspondientes a la variable y sus probabilidades y se calculan la media y la varianza teóricas sustituyendo en las fórmulas anteriores:

(X1 <- c(0:3))

## [1] 0 1 2 3

(p <- c(64/125, 48/125, 12/125, 1/125))

## [1] 0.512 0.384 0.096 0.008

(mu <- sum(X1\*p)) # media teórica

## [1] 0.6

(sigma2 <- sum((X1-mu)^2\*p)) # varianza teórica

## [1] 0.48

**Apartado 2:** Combina sample con replicate para simular cien mil muestras de tamaño 10 de esta variable . 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.

**Respuesta:**

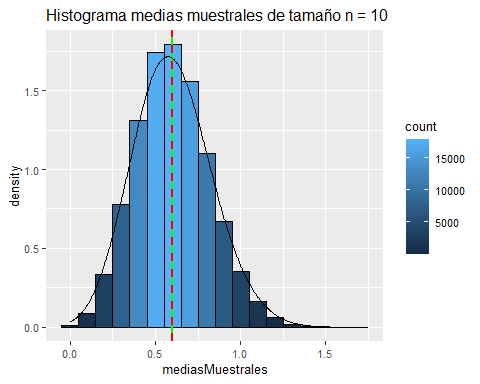
Generación de cien mil muestras de tamaño 10 calculando su media:

set.seed(2021)  
k = 100000 # nº muestras  
n = 10 # tamaño de cada muestra  
mediasMuestrales = replicate(k, {  
muestra = sample(0:3, size = n, replace = TRUE, prob = c(64, 48, 12, 1))  
mean(muestra)  
})  
  
head(mediasMuestrales, 10) # 10 primeras medias muestrales

## [1] 0.8 0.8 1.1 1.1 0.6 0.5 0.3 0.7 0.7 1.0

Representación mediante un histograma de la distribución de medias junto con la media teórica presentada mediante línea discontinua:

# Histograma con densidad  
mediasMuestrales\_df <- as.data.frame(mediasMuestrales)  
g1 <- ggplot(mediasMuestrales\_df, aes(x = mediasMuestrales)) +  
 geom\_histogram(binwidth=0.1,aes(y = ..density.., fill=..count..),  
 colour = 1) +  
 geom\_vline(aes(xintercept = mean(mediasMuestrales)), colour = "green", size = 1)+  
 geom\_vline(aes(xintercept = mu), linetype = "dashed", colour = "red", size = 1)+  
 geom\_density(adjust = 4) # adjust suaviza la línea de densidad  
  
g1+  
 theme (text = element\_text(size=10)) + # Tamaño de fuente del grafico por defecto  
 ggtitle ("Histograma medias muestrales de tamaño n = 10") # Título del gráfico



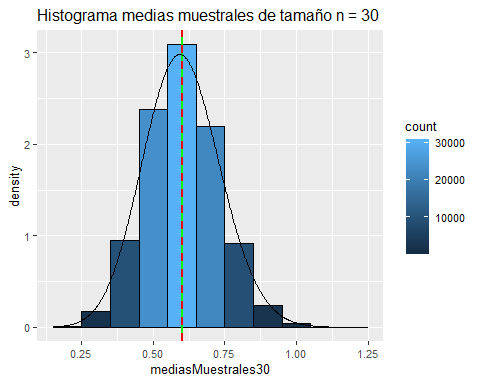
En el histograma se representa la distribución de las medias muestrales y, se puede ver como la mayoría de medias se concentra muy próxima a la media teórica. Además, si simultáneamente se dibuja la línea con la media de las medias, ésta se posiciona encima de la teórica coincidiendo con ella.

Si cambiamos el tamaño de la muestra a 30:

set.seed(2021)  
k = 100000 # nº muestras  
mediasMuestrales30 = replicate(k, {  
muestra = sample(0:3, size = 30, replace = TRUE, prob = c(64, 48, 12, 1))  
mean(muestra)  
})  
  
head(mediasMuestrales30, 10) # 10 primeras medias muestrales

## [1] 0.9000000 0.7333333 0.5666667 0.5333333 0.5333333 0.4000000 0.3666667  
## [8] 0.6000000 0.6000000 0.4666667

mediasMuestrales\_df <- as.data.frame(mediasMuestrales30)  
  
g2 <- ggplot(mediasMuestrales\_df, aes(x = mediasMuestrales30)) +  
 geom\_histogram(binwidth=0.1,aes(y = ..density.., fill=..count..),  
 colour = 1) +  
 geom\_vline(aes(xintercept = mean(mediasMuestrales30)), colour = "green", size = 1)+  
 geom\_vline(aes(xintercept = mu), linetype = "dashed", colour = "red", size = 1)+  
 geom\_density(adjust = 4) # adjust suaviza la línea de densidad  
  
g2+  
 theme (text = element\_text(size=10)) + # Tamaño de fuente del grafico por defecto  
 ggtitle ("Histograma medias muestrales de tamaño n = 30") # Título del gráfico



Con cien mil muestras de tamaño 30, el rango para los valores de las medias se ha hecho más estrecho y, al igual que ocurría con tamaño 10, la media de las medias de las muestras coincide con la media teórica.

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

$$
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{valor de }X\_2 & 0 & 1 & 2 \\
\hline
\text{Probabilidad de ese valor }P(X = x\_i) & \dfrac{1}{2} &
\dfrac{1}{4}& \dfrac{1}{4}\rule{0mm}{6mm} \\[3mm]
\hline
\end{array}
$$

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

**Respuesta:**

Creamos una variable y calculamos sus probabilidades asociadas que, dado que son independientes, el resultado es el producto es decir:

# Vector de probabilidades para cada combinación posible  
(p <- c(64/125, 48/125, 12/125, 1/125) \* rep(c(1/2, 1/4, 1/4), each = 4))

## [1] 0.256 0.192 0.048 0.004 0.128 0.096 0.024 0.002 0.128 0.096 0.024 0.002

X1 <- rep(0:3,3)  
X2 <- rep(0:2, each = 4)  
  
# Tabla de probabilidad  
(table <- data.frame(Z=X1 + X2, X1, X2, p))

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

# Tabla de probabilidad de la variable suma  
(table2 <- table %>%   
 group\_by(Z) %>%   
 summarise(p = sum(p)))

## # A tibble: 6 x 2  
## Z p  
## <int> <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

La variable suma designada como Z puede tomar 6 posibles valores (del 0 al 5) y sus probabilidades se calculan en función de la probabilidad obtenida para cada combinación. Por ejemplo:

Así también podríamos ver que la suma de las probabilidades para esta nueva variable cumple el axioma sumando 1:

sum(table2$p)

## [1] 1

**Apartado 4:** Calcula la media teórica de la suma . 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.

**Respuesta:**

Para calcular la media teórica de las suma, sabemos que:

# Variable X1  
(X1 <- c(0:3))

## [1] 0 1 2 3

# Probabilidades para X1  
(p <- c(64/125, 48/125, 12/125, 1/125))

## [1] 0.512 0.384 0.096 0.008

# Variable X2  
(X2 <- c(0:2))

## [1] 0 1 2

# Probabilidades para X2  
(p2 <- c(1/2, 1/4, 1/4))

## [1] 0.50 0.25 0.25

# Media teórica variable Z = X1 + X2  
# Forma 1: a partir de las medias de las variables aleatorias  
(mu\_z <- sum(X1\*p) + sum(X2\*p2))

## [1] 1.35

# Forma 2: a partir de la nueva variable  
sum(table2$Z\*table2$p)

## [1] 1.35

Simular cien mil valores variable suma: como son independientes, se simula cada variable aleatoria por separado y se calcula su media

set.seed(2021)  
k = 100000 # nº muestras  
mediasMuestrales = replicate(k, {  
X1 = sample(0:3, size = 1, replace = TRUE, prob = c(64, 48, 12, 1))  
X2 = sample(0:2, size = 1, replace = TRUE, prob = c(1/2, 1/4, 1/4))  
mean(X1)+mean(X2)  
})

O bien se toman muestras de tamaño 1 para la nueva variable Z:

set.seed(2021)  
k = 100000 # nº muestras  
mediasMuestrales\_Z = replicate(k, {  
Z = sample(0:5, size = 1, replace = TRUE, prob =table2$p)  
mean(Z)  
})

Si comparamos los tres resultados, vemos que son similares:

tibble( medias\_Z = mean(mediasMuestrales\_Z),  
medias\_X1\_X2 = mean(mediasMuestrales),  
mu\_z = mu\_z)

## # A tibble: 1 x 3  
## medias\_Z medias\_X1\_X2 mu\_z  
## <dbl> <dbl> <dbl>  
## 1 1.35 1.35 1.35

# 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*.

**Respuesta:**

En primer lugar, leemos los datos, vemos sus 6 primeras líneas y el nombre de cada una de las variables:

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 conjunto de datos se comopone de un total de 9 columnas, sin embargo, una de las razones por las que este dataset no cumple los principios de tidy data es la combinación de las variables sexo y edad en una sola columna llamada “gender\_age”. Por ese motivo, primeramente, separamos en dos dicha columna:

testResults <- testResults %>% separate (gender\_age, c("gender", "age"))  
  
testResults$gender <- as.factor(testResults$gender) # convertimos sexo en factor  
testResults$age <- as.numeric(testResults$age) # convertimos edad en numérico  
head(testResults)

## # A tibble: 6 x 10  
## name id gender age test\_number week1 week2 week3 week4 week5  
## <chr> <dbl> <fct> <dbl> <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

El segundo motivo por el que testResults no es tidy data es debido a la variable “week”. Cada semana, los alumnos realizaron dos tipos de test pero en lugar de tener una variable que recoja la semana en la que se ha hecho la prueba y otra variable con la puntuación obtenida, nos encontramos con 5 columnas correspondientes a la puntuación de los test en cada semana. Así, se realiza el cambio de formato ancho a longitudinal y, además, se convierte la variable week en numérico para poder tratar dicha variable como factor o numérica según el objetivo de estudio que se plantee:

testResults\_Tidy = testResults %>%  
 pivot\_longer(week1:week5, names\_to = "week") %>%   
 separate(week, c("aux", "week"),convert=TRUE, sep = 4) %>%   
 select(-aux)  
   
  
head(testResults\_Tidy)

## # A tibble: 6 x 7  
## name id gender age test\_number week value  
## <chr> <dbl> <fct> <dbl> <dbl> <int> <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

# 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](https://r4ds.had.co.nz/exploratory-data-analysis.html#exercises-17). Las ideas de esa sección son importantes para nuestro trabajo de las próximas sesiones.

**Respuesta:**

Leemos el enunciado: 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?

Cargamos en memoria el dataset:

data("diamonds")  
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

dim(diamonds)

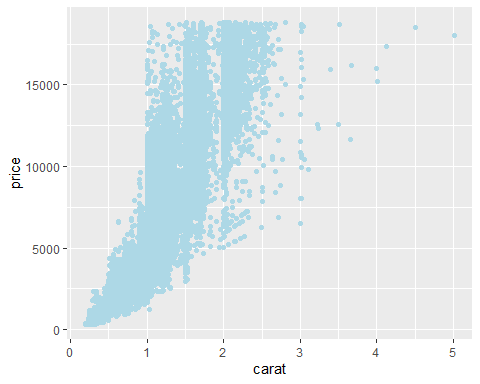
## [1] 53940 10

Las variables a estudiar en este análisis son: carat (quilates del diamante), cut (calidad del corte), color (color del diamante), clarity (cómo de limpio está el diamante), depth (porcentaje de profundidad teniendo en cuenta la longitud x, la anchura y y la profundidad z), table (anchura de la parte superior del diamente desde su punto más ancho). Las variables x, y, z, al estar incluidas dentro de depth se omiten.

De las variables mencionadas anteriormente, carat y depth son variables continuas, por tanto, como la variable respuesta (price) también lo es, se realiza un gráfico de dispersión para analizar su tendencia:

* Carat:

ggplot(diamonds, aes(x = carat, y = price)) +  
 geom\_point(colour = "lightblue")



Para los quilates del diamante vemos que, a medida que aumenta la cifra para esta variable, también lo hace el precio.

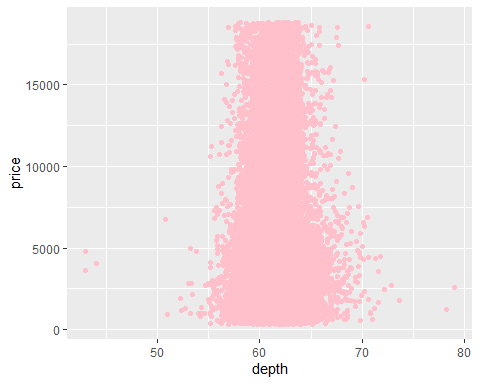
Además, si calculamos la correlación de pearson entre estas dos variables, su valor es bastante próximo a 1 y positivo, lo que quiere decir que, a más quilates mayor precio.

cor(diamonds$carat,diamonds$price)

## [1] 0.9215913

* Depth:

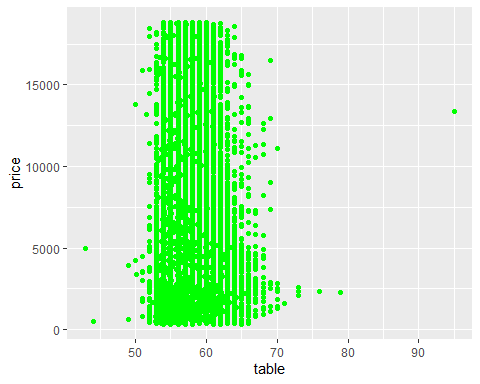
ggplot(diamonds, aes(x = depth, y = price)) +  
 geom\_point(colour = "pink")



En el diagrama de dispersión, se aprecia que dentro del rango de la profundidad, la variabilidad de los precios es muy amplia variando desde los más bajos a los más altos por lo que no sería un buen predictor del precio.

* Table:

ggplot(diamonds, aes(x = table, y = price)) +  
 geom\_point(colour = "green")

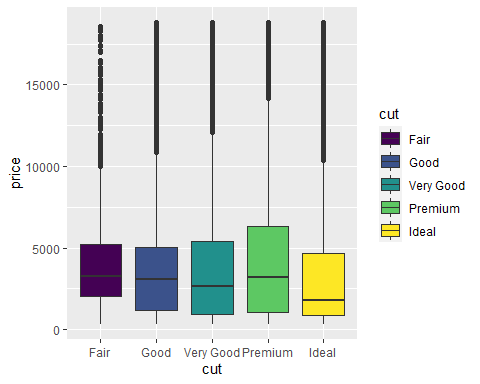


Para la variable table, no parece existir relación con el precio puesto que no se observa ninguna tendencia en el diagrama de dispersión.

Y para las variables categóricas, se representan los boxplots:

* Cut:

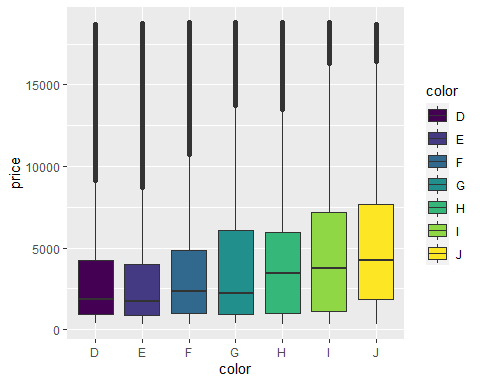
ggplot(data = diamonds) +  
 geom\_boxplot(mapping = aes(x = cut, y = price, fill = cut))



En el corte, existe una gran variabilidad dentro de cada una de las categorías pero muy poca entre ellas.

* Color:

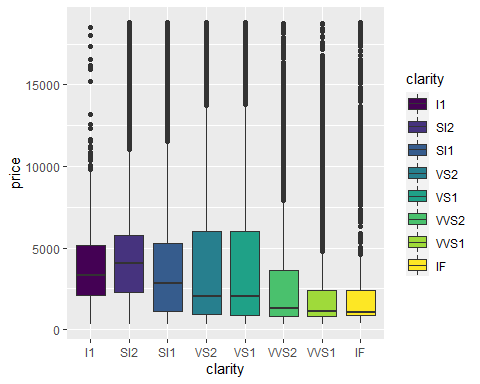
ggplot(data = diamonds) +  
 geom\_boxplot(mapping = aes(x = color, y = price, fill = color))



En el color, aunque la variabilidad es bastante elevada en todas las categorías, se puede observar cierta tendencia: cuanto peor es (J), mayor variabilidad hay.

* Clarity:

ggplot(data = diamonds) +  
 geom\_boxplot(mapping = aes(x = clarity, y = price, fill = clarity))

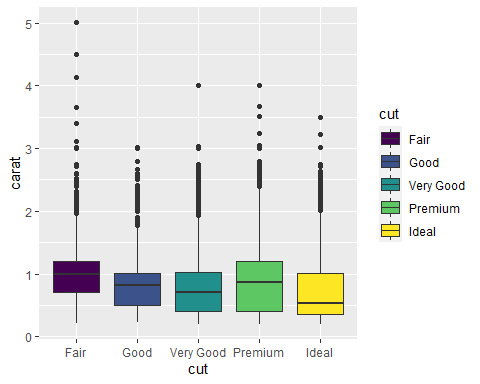


En la limpieza, la variabilildad del precio parece ser mayor para los que menos limpios son, no obstante, en todas las categorías la variabilidad es bastante amplia.

Respondiendo a la pregunta del ejercicio, la variable “carat” es la que parece predecir mejor el precio del diamante. Continuando con la siguieste cuestión, estudiamos la relación de esta variable con el corte (“cut”).

Relación de carat con cut:

ggplot(data = diamonds) +  
 geom\_boxplot(mapping = aes(x = cut, y = carat, fill = cut))



Mediante este boxplot se observa que para el peso del diamante existe una gran variabilidad dentro de cada categoría del corte. Además, fijándonos en la primera categoría (Fair) esta alcanza valores ligeramente más elevados que las otras, por lo que, respondiendo a la última pregunta, como para la categoría Fair se alcanzan valores más altos para carat, esto implicaría también un precio más alto (según lo observado en el diagrama de dispersión de carat con price).

* Haz el [ejercicio 4 de la Sección 12.6.1 de R4DS](https://r4ds.had.co.nz/tidy-data.html#exercises-27). ¡Aprovecha el código previo de esa sección para trabajar con datos limpios!

**Respuesta:**

For each country, year, and sex compute the total number of cases of TB. Make an informative visualisation of the data.

Para este apartado, se emplea la base de datos who, la cual contiene información sobre casos de tuberculosis

data(who) # cargamos en memoria el dataset  
head(who) # 6 primeras filas

## # A tibble: 6 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  
## # ... 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>, ...

Seguimos los pasos del capítulo para convertir el dataset en tidy data:

* Las variables desde new\_sp\_m014 a newrel\_f65 serán transformadas a un formato longitudinal, omitiendo los valores missing y denotando a esa nueva variable como key
* Separamos la columna key
* Eliminamos columnas redundantes y separamos la columna sex\_age en dos

who <- 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(who)

## # 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

Calculamos el nº total de casos de tuberculosis por país, año y sexo:

* Representación mediante tabla:

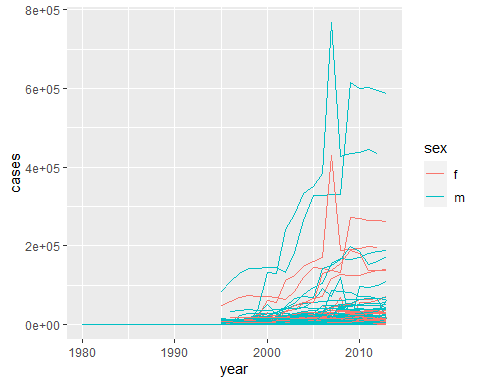
who %>%  
 group\_by(country, year, sex) %>%  
 summarise(cases = sum(cases))

## `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

Representación mediante gráfico de espaguetis:

who %>%  
 group\_by(country, year, sex) %>%  
 summarise(cases = sum(cases)) %>%  
 unite(country\_sex, country, sex, remove = FALSE) %>%  
 ggplot(aes(x = year, y = cases, group = country\_sex, colour = sex)) +  
 geom\_line()



El nº de casos hasta 1995 era muy bajo en comparación a los demás años y lo mismo ocurre con algunos países:

who %>%  
 group\_by(year) %>%  
 summarise(cases = sum(cases))

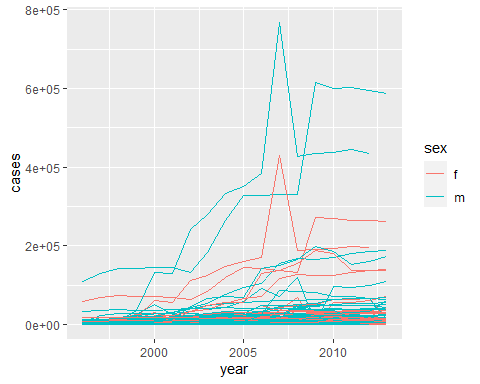
## # A tibble: 34 x 2  
## year cases  
## <int> <int>  
## 1 1980 959  
## 2 1981 805  
## 3 1982 824  
## 4 1983 786  
## 5 1984 814  
## 6 1985 799  
## 7 1986 754  
## 8 1987 670  
## 9 1988 682  
## 10 1989 654  
## # ... with 24 more rows

who %>%  
 group\_by(country) %>%  
 summarise(cases = sum(cases)) %>%   
 arrange(cases)

## # A tibble: 219 x 2  
## country cases  
## <chr> <int>  
## 1 Bonaire, Saint Eustatius and Saba 0  
## 2 Tokelau 0  
## 3 Anguilla 2  
## 4 San Marino 2  
## 5 Monaco 3  
## 6 Niue 3  
## 7 Montserrat 4  
## 8 British Virgin Islands 5  
## 9 US Virgin Islands 7  
## 10 Bermuda 8  
## # ... with 209 more rows

Por eso, para una mejor visualización, representamos los años a partir de 1995:

who %>%  
 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()



Con este gráfico vemos la evolución a lo largo de los años para todos los países y según el sexo. Sin embargo, cuesta diferenciar cuál es la línea correspondiente a cada país. Por este motivo, podemos hacer un gráfico para cada territorio pero previamente a esta representación, se van a seleccionar los países con un número de casos mayor a la media :

# Creamos una tabla con los países que tienen un nº de casos mayor a la media  
tabla <- who %>%  
 group\_by(country) %>%  
 summarise(cases = sum(cases)) %>%   
 arrange(cases) %>%   
 filter(cases > mean(cases))  
  
  
  
 who %>%  
 group\_by(country, year, sex) %>%  
 filter(year>1995 & country %in% tabla$country) %>%   
 summarise(cases = sum(cases)) %>%  
 ggplot(aes(x = year, y = cases, group = sex, colour = sex)) +  
 geom\_line()+  
 facet\_wrap(~country) +  
 theme(axis.text.x = element\_text(size= rel(0.6)))

