Práctica 1. FMAD 2021-2022

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

Monsalve Rodilla, Ignacio

Curso 2021-22. Última actualización: 2021-09-15

Contenido

EJERCICIO I

Apartado I

```
library(tidyverse)
```

Se ofrece una tabla de densidad de probabilidad de la variable aleatoria X1.

Creamos un vector llamado valor y otro llamado probabilidad

```
valor = c(0,1,2,3)

prob = c(64/125, 48/125, 12/125, 1/125)
```

La media se calcula simplemente multiplicando los valores por sus respectivas probabilidades:

```
media = valor %*% prob
(media = media[1,1])

## [1] 0.6
```

La varianza teórica se puede calcular de la siguiente forma:

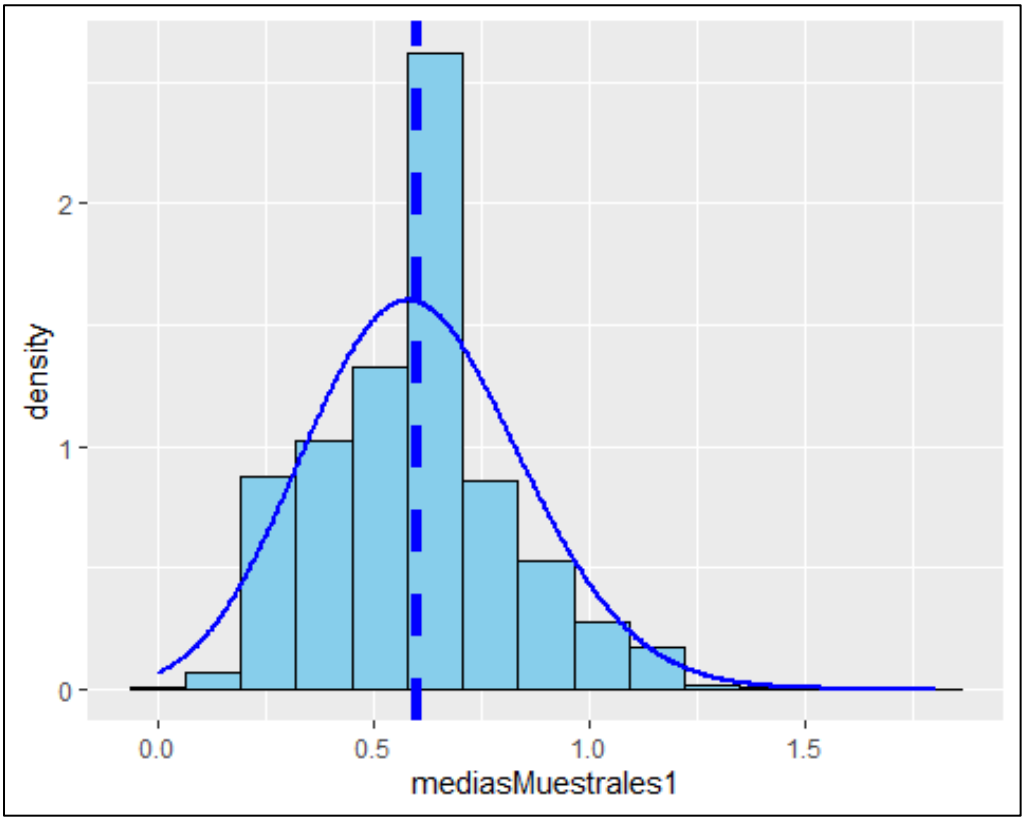
```
sum((valor-media)^2*prob)

## [1] 0.48
```

Apartado II

```
valor = c(0,1,2,3)

prob = c(64/125, 48/125, 12/125, 1/125)
```

Se combinan 'sample' y 'replicate' para simular 100.000 muestras de tamaño 10 y estudiar la distribución de las medias muestrales.

```
k = 100000
n1 = 10
mediasMuestrales1 = replicate (k, {
  muestra = sample(valor, n1, replace = TRUE, prob = prob)
  mean(muestra)
})

ggplot() +
  geom_histogram(aes(x = mediasMuestrales1, y=..density..), bins = 15, fi
ll="skyblue", color="black") +
  geom_vline(xintercept = mean(mediasMuestrales1),
             col="blue", linetype="dashed", size=2)+
  geom_density(mapping = aes(mediasMuestrales1), adjust=6,color="blue",si
ze=1, adjust=1.5)
```

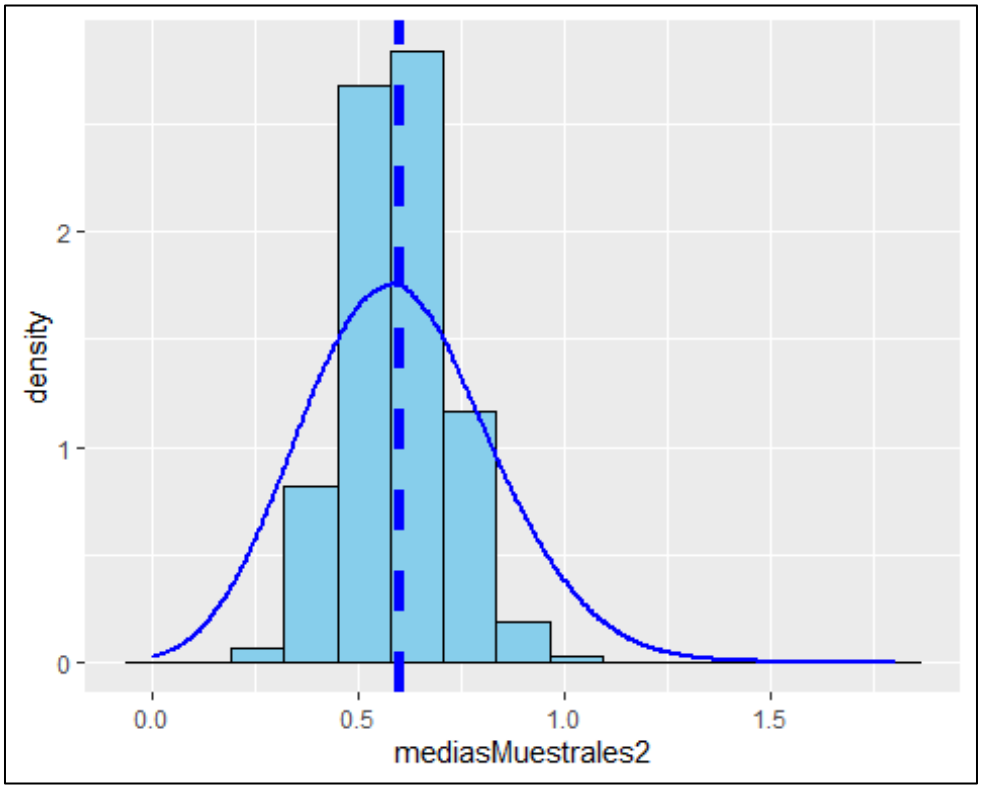


Se repite con tamaño 30 en vez de 10, por lo que es de esperar que la acentuación del histograma hacia la normal sea mayor que antes.

```
k = 100000
n1 = 30
mediasMuestrales2 = replicate (k, {
  muestra = sample(valor, n1, replace = TRUE, prob = prob)
  mean(muestra)
})
```

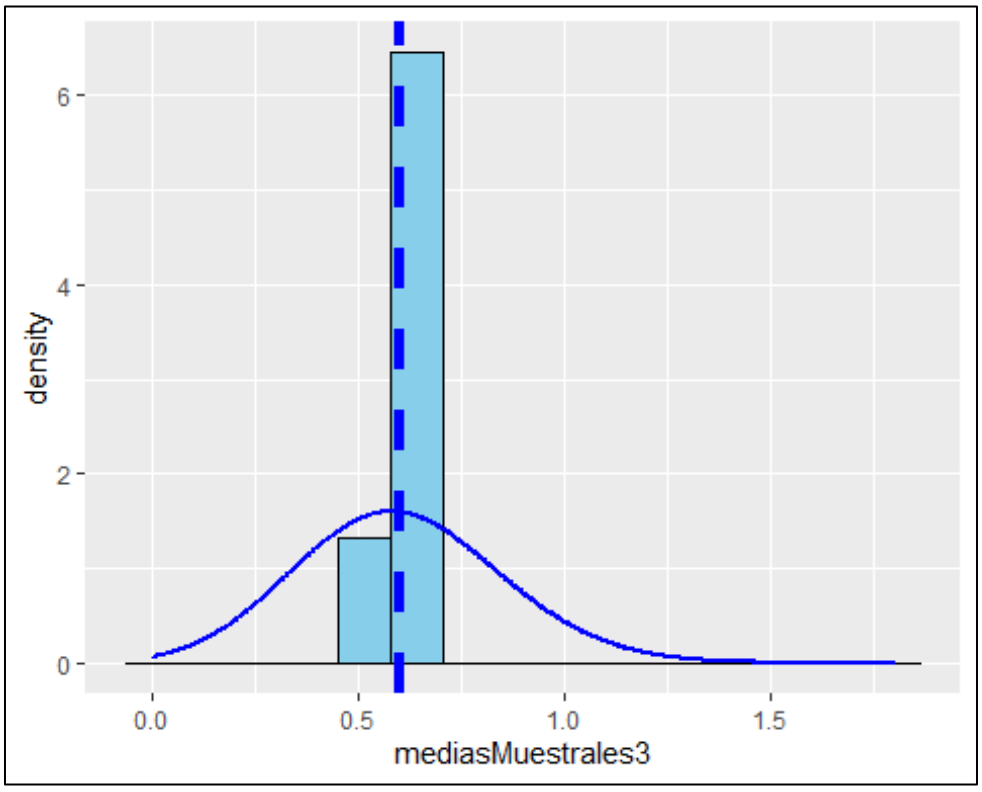
De nuevo, se puede representar con ggplot:

```
ggplot() +
  geom_histogram(aes(x = mediasMuestrales2, y=..density..), bins = 15, fi
ll="skyblue", color="black") +
  geom_vline(xintercept = mean(mediasMuestrales1),
             col="blue", linetype="dashed", size=2)+
  geom_density(mapping = aes(mediasMuestrales1), adjust=3,color="blue",si
ze=1, adjust=1.5)
```



No se pide, pero por reforzar la idea, se incluye también un tamaño de muestra igual a 1000.

```
k = 10000
n1 = 1000
mediasMuestrales3 = replicate (k, {
  muestra = sample(valor, n1, replace = TRUE, prob = prob)
  mean(muestra)
})

ggplot() +
  geom_histogram(aes(x = mediasMuestrales3, y=..density..), bins = 15, fi
ll="skyblue", color="black") +
  geom_vline(xintercept = mean(mediasMuestrales1),
             col="blue", linetype="dashed", size=2)+
  geom_density(mapping = aes(mediasMuestrales1), adjust=6,color="blue",si
ze=1, adjust=1.5)
```



Apartado III

La variable aleatoria discreta X2 tiene otra tabla de probabilidad, por lo que se guardan dos nuevos vectores, nombrados como valor2 y prob2, y también se vuelven a incluir los del apartado I ya que se necesitarán y está bien recordarlos.

```
valor = c(0,1,2,3)

prob = c(64/125, 48/125, 12/125, 1/125)




valor2 = c(0,1,2)

prob2 = c(1/2, 1/4, 1/4)
```

Suponiendo que X1 y X2 son independientes:

- **¿Qué valores puede tomar la suma X1+X2?**

  Los valores que puede tomar son los comprendidos entre 0 y 5

- **¿Cuál es tu tabla de probabilidad?**

  Para la tabla de probabilidad se utilizan bucles for y se multiplican las distintas probabilidades de X1 y X2 para cada valor que pueden tomar.

```
probf <- vector(length=6)
for(i in seq(0,5,1)){
  for(j in seq(0,3,1)){
    for(k in seq(0,2,1)){
      if((j+k)==i){
        probf[i+1] <- probf[i+1] + (prob[j+1]*prob2[k+1])
      }
    }
  }
}
probf

## [1] 0.256 0.320 0.272 0.124 0.026 0.002

sum(probf)

## [1] 1
```

Apartado IV

La media teórica de la suma se calcula:

```
media_ej_tot = valor%*%prob + valor2%*%prob2
media_ej_tot[1,1]

## [1] 1.35
```

También podría hacerse de esta otra forma, es lo mismo.

```
a = seq(0,5,1)
a

## [1] 0 1 2 3 4 5

probf

## [1] 0.256 0.320 0.272 0.124 0.026 0.002

media_ej = a%*%probf
media_ej[1,1]

## [1] 1.35
```

A continuación, se simulan cien mil valores:

```
k = 100000
mediasMuestrales4 = 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)
})

(medias_ambas_variables = mean(mediasMuestrales4))

## [1] 1.34359
```

Ejercicio II

En primer lugar, se descarga el fichero del enlace y se guarda en una variable llamada 'datos2'

```
library(tidyverse)

datos2 = read_csv('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 th
is message.

View(datos2)
```

Se puede ver como la información de gender y age se encuentra en la misma tabla, separada por un _, por lo que no se cumplen los principios 'tidy'. Lo que se quiere conseguir es de esa columna, obtener dos columnas en las que se pueda diferenciar por una parte la variable 'gender' y por otra 'age'. Esto se realiza con separate.

```
datosLimpios <- datos2 %>%
                separate(gender_age, into = c("gender", "age"), sep = "_"
, convert = TRUE) %>%
                pivot_longer(c("week1", "week2", "week3", "week4", "week5
"), names_to = "Type of week",
                             values_to = "value")
```

También se utiliza "pivot_longer" ya que las distintas semanas aparecen en diferentes columnas, y nos gustaría tener todo ello en una única columna que se nombra como 'Type of week'

```
View(datosLimpios)
```

Ejercicio III

Apartado a) (Sección 7.5.1.1)

- **¿Qué variables en el dataset Diamonds son las más importantes para predecir el precio del diamante?**

En primer lugar, se deberían escoger aquellas variables que pueden ser posibles candidatas.

Como primer análisis exploratorio, se mirará por encima el dataset.

```
View(diamonds)
```

En este caso existe un número de columnas razonable, y será más o menos sencillo limitar la búsqueda, sin embargo, en el mundo real podemos encontrar datasets con muchas más columnas y será también importante el conocimiento que se tenga sobre el tema para poder descartar algunas variables que se sepa que no influyen tanto y ahorrarse mucho tiempo y coste computacional.

También puede ocurrir que no se sepa qué es exactamente cada columna, y una de las ventajas de trabajar con este tipo de datasets es que está muy bien organizado y estandarizado, por lo que si se utiliza el comando diamonds, nos dará una breve descripción del asunto.

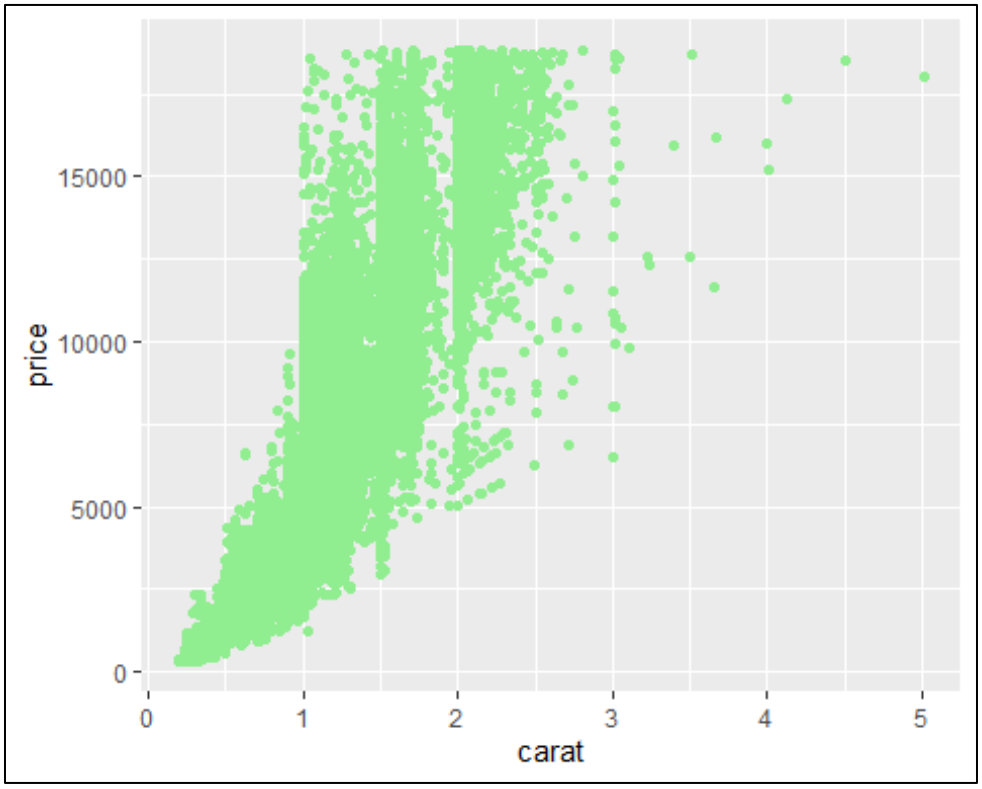
```
diamonds

## starting httpd help server ... done
```

Se puede ver una pequeña descripción en la que se incluye, por ejemplo: carat: peso del diamante cut: calidad del diamante desde 'Normal' hasta 'Ideal' color: una escala en letras desde D (lo mejor) hasta J (lo peor) clarity: mide cómo de transparente es x, y, z: son coordinadas, que de alguna forma se puede interpretar que se encuentran "agrupadas" dentro de la variable 'carat'

En primer lugar, se debe realizar una distinción entre variables continuas y otro tipo de variables como pueden ser factores.

Para la variable carat, por ejemplo, que es el peso del diamante, se puede realizar un gráfico de puntos:

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



Para terminar de cuantificar el análisis, podría realizarse una **regresión lineal**.

```
regresion <- lm(carat ~ price, data = diamonds)
summary(regresion)
```

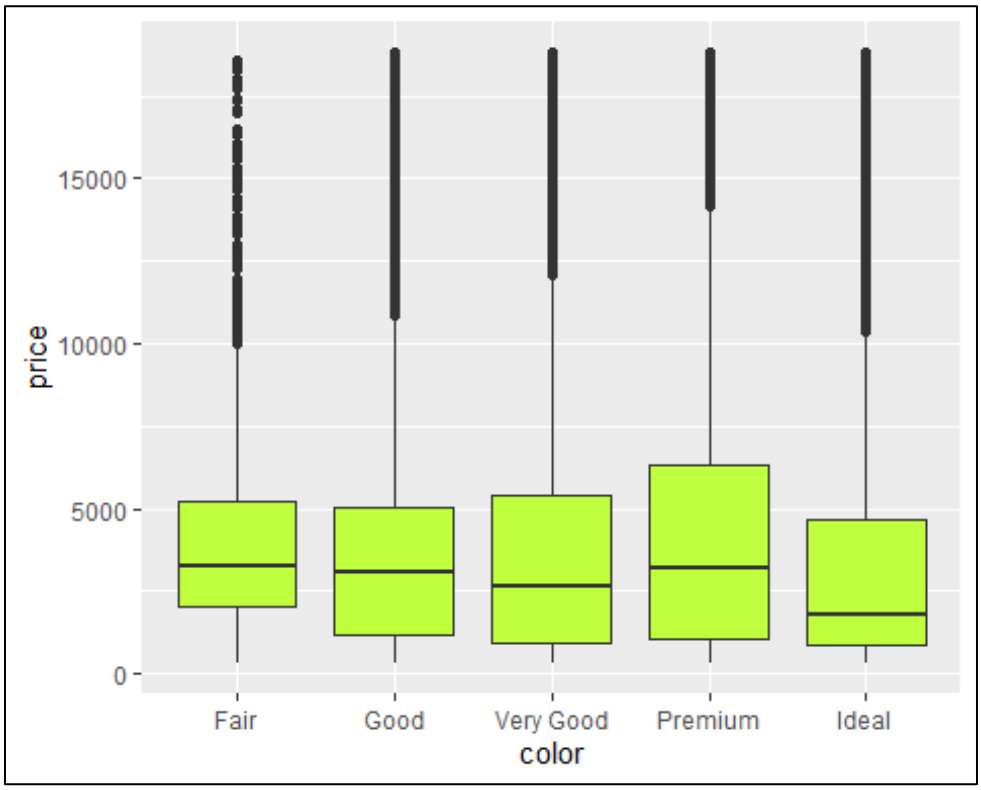
```
##
## Call:
## lm(formula = carat ~ price, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.35765 -0.11329 -0.02442  0.10344  2.66973
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.673e-01  1.112e-03   330.2   <2e-16 ***
## price       1.095e-04  1.986e-07   551.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.184 on 53938 degrees of freedom
## Multiple R-squared:  0.8493, Adjusted R-squared:  0.8493
## F-statistic: 3.041e+05 on 1 and 53938 DF,  p-value: < 2.2e-16
```

Se puede ver como el **R^2 = 85%,** que es bastante elevado.

Los análisis también se pueden realizar integrando un boxplot como se explica en R for Data Scientist.

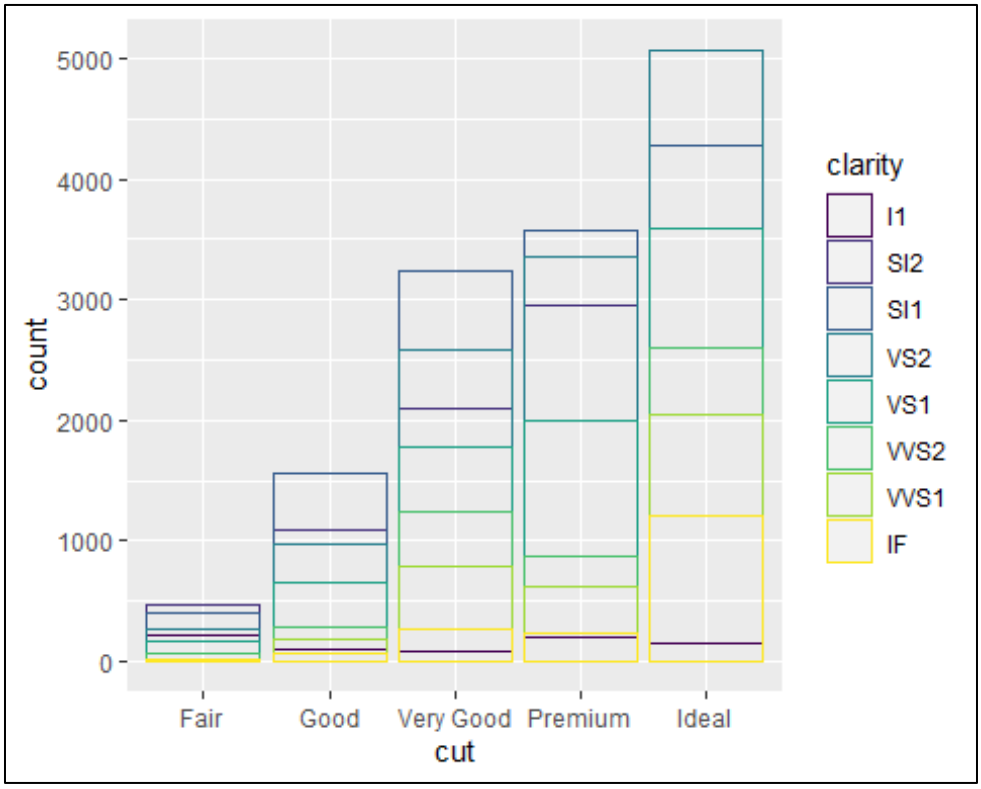
Para mostrar esta opción, se analizará la variable 'color' con los boxplot, de forma que el gráfico resulta:

```
diamonds %>%
  mutate(color = fct_rev(color)) %>%
  ggplot(aes(x = color, y = price)) +
  geom_boxplot(mapping = aes(x = cut, y = price), fill = 'olivedrab1')
```



En el caso de la variable 'clarity' ocurre algo similar, en este caso va a representarse con un gráfico distinto:

```
ggplot(data = diamonds, mapping = aes(x = cut, colour = clarity)) +
  geom_bar(fill = NA, position = "identity")
```



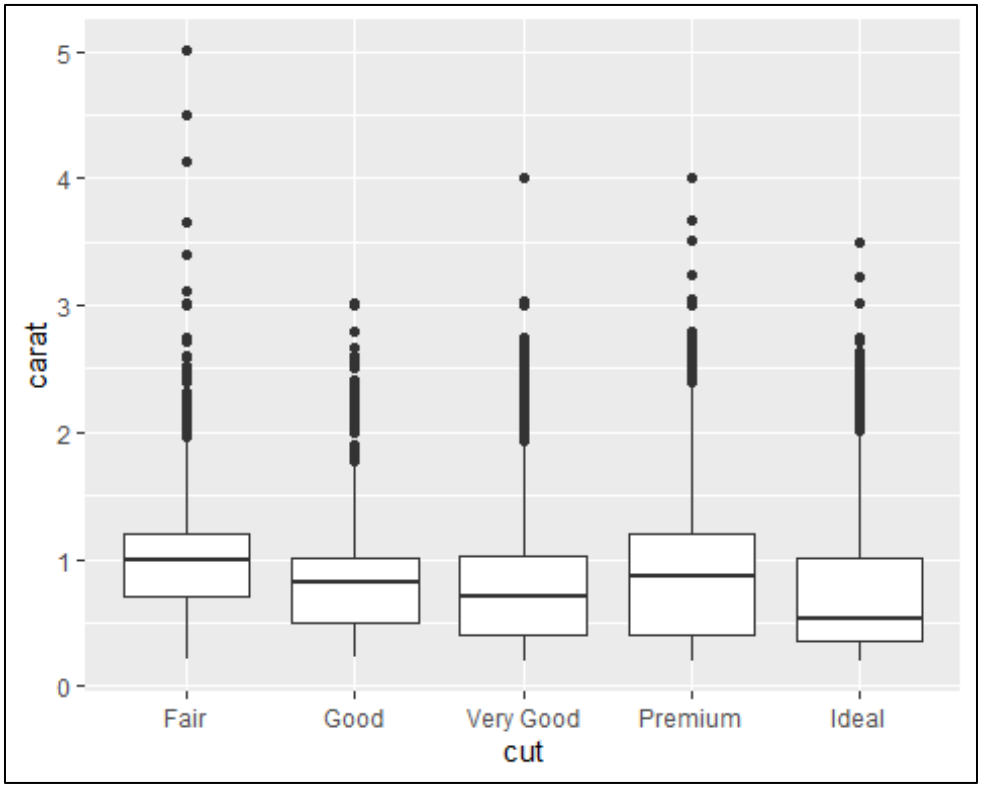
En este caso, el mejor predictor sería la variable carat.

Por ir un paso más allá, se puede realizar también una regresión lineal para cuantificar de alguna forma cómo de bueno sería el modelo.

- **¿Cómo está dicha variable relacionada con 'cut'?**

En esta segunda pregunta, la relación entre 'cut' y 'carat' se puede representar de la siguiente forma:

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



- **¿Por qué la combinación de las dos relaciones hace que los diamantes de peor calidad sean más caros?**

En el gráfico expuesto anteriormente puede verse que existe mucha variabilidad dentro de cada categoría. Se puede ver como en la primera categoría, los valores son más elevados que en resto, lo cual explica el hecho de que el precio sea más caro.

Apartado b) (Sección 12.6.1)

Para cado país, año y sexo, computar el número total de casos de TB. Realizar también un gráfico explicativo.

En primer lugar, se van a ver los datos.

```
library(tidyr)

data(who)
head(who, 5)

## # A tibble: 15 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



library(tidyverse)
```

A continuación, se deben realizar algunos pasos para limpiar y ordenar la tabla

```
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)
```

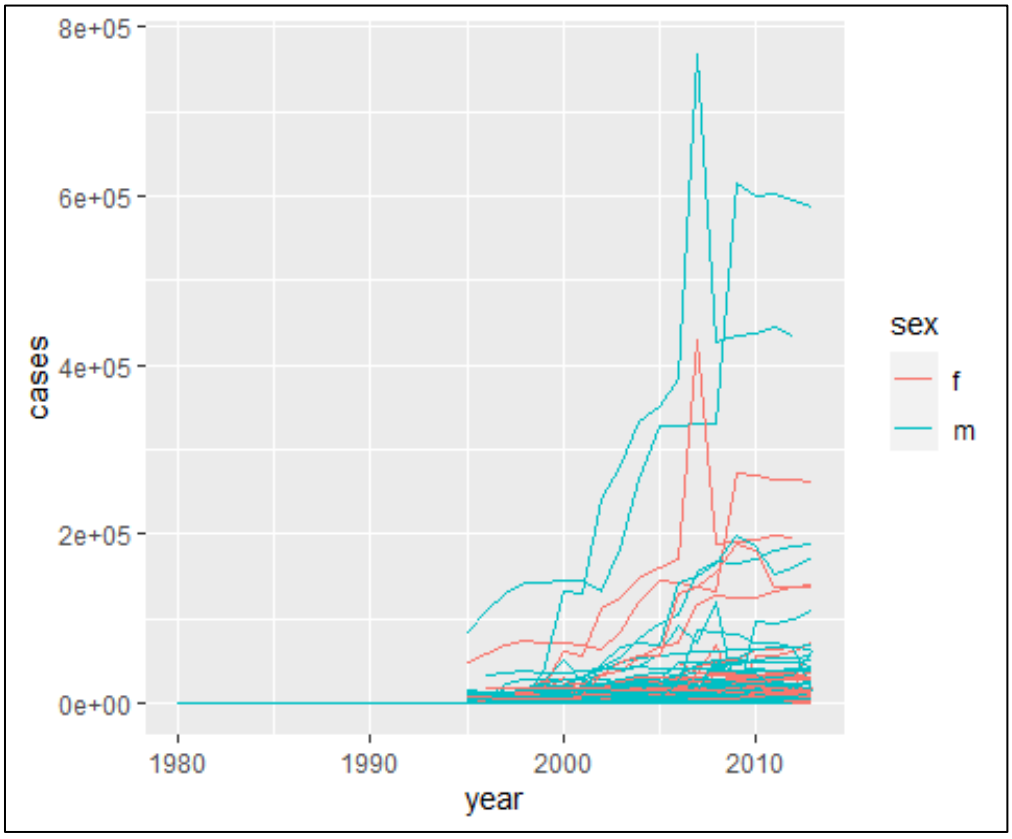
Para calcular el número total de casos de tuberculosis:

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

```
## # A tibble: 6,921 x 4
## # Groups:   country, year [3,484]
##    country      year sex   cases
##    <chr>       <int> <chr> <int>
##  1 Afghanistan  1997 f       102
##  2 Afghanistan  1997 m        26
##  3 Afghanistan  1998 f      1207
##  4 Afghanistan  1998 m       571
##  5 Afghanistan  1999 f       517
```

```
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()
```

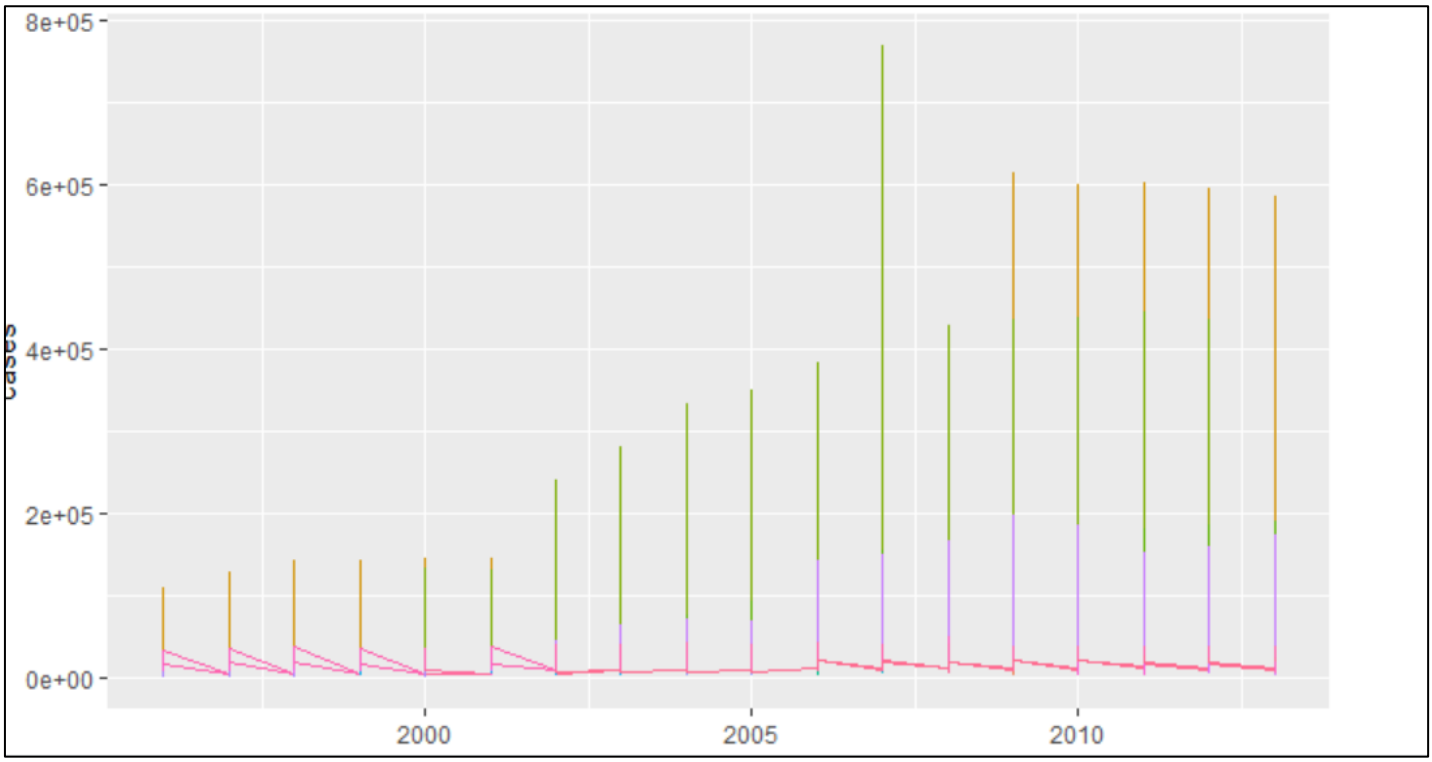
```
## `summarise()` has grouped output by 'country', 'year'. You can overrid
e using the `.groups` argument.
```



```r
tabla <- who %>%
  group_by(country) %>%
  summarise(cases = sum(cases)) %>%
  arrange(cases) %>%
  filter(cases > mean(cases))

View(who)
```

Un posible gráfico podría ser este:

```r
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 = country)) +
  geom_line()
```



Cuya leyenda sería:

Otra opción sería utilizar los gráficos desglosados de diferente forma, como por ejemplo por países, y utilizar una escala diferente para cada país para poder visualizarlo de una forma correcta.