Tarea 2

Álvaro Francisco Ruiz Cornejo

18/9/2021

Ejercicio 1. Simulando variables aleatorias discretas.

- **Apartado 1: La variable aleatoria discreta X1 tiene esta tabla de densidad de probabilidad (es la variable que se usa como ejemplo en la Sesión). Calcula la media y la varianza teóricas de esta variable.**

valor de $X1$	0	1	2	3
Probabilidad de ese valor $P(X = x_i)$	$\frac{64}{125}$	$\frac{48}{125}$	$\frac{12}{125}$	$\frac{1}{125}$

Convertimos la tabla a información legible por R:

```
xi = c(0,1,2,3)
pi = c(64/125, 48/125, 12/125, 1/125)
```

Calculamos la media teórica, multiplicando cada valor de x1 por su correspondiente probabilidad:

```
(media = as.numeric(xi %*% pi)) # Producto escalar
```

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

De igual manera, calculamos la varianza teórica de la siguiente manera:

```
(varianza = sum((xi - media)^2 * pi))
```

```
## [1] 0.48
```

- **Apartado 2: Combina sample con replicate para simular cien mil muestras de tamaño 10 de esta variable X1. 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.**

```
k = 100000
poblacion = 0:3

mediasMuestrales = replicate(k, {
  muestra = sample(poblacion, size = 30, replace = TRUE, prob = c(64, 48, 12, 1))
  mean(muestra)
})
```

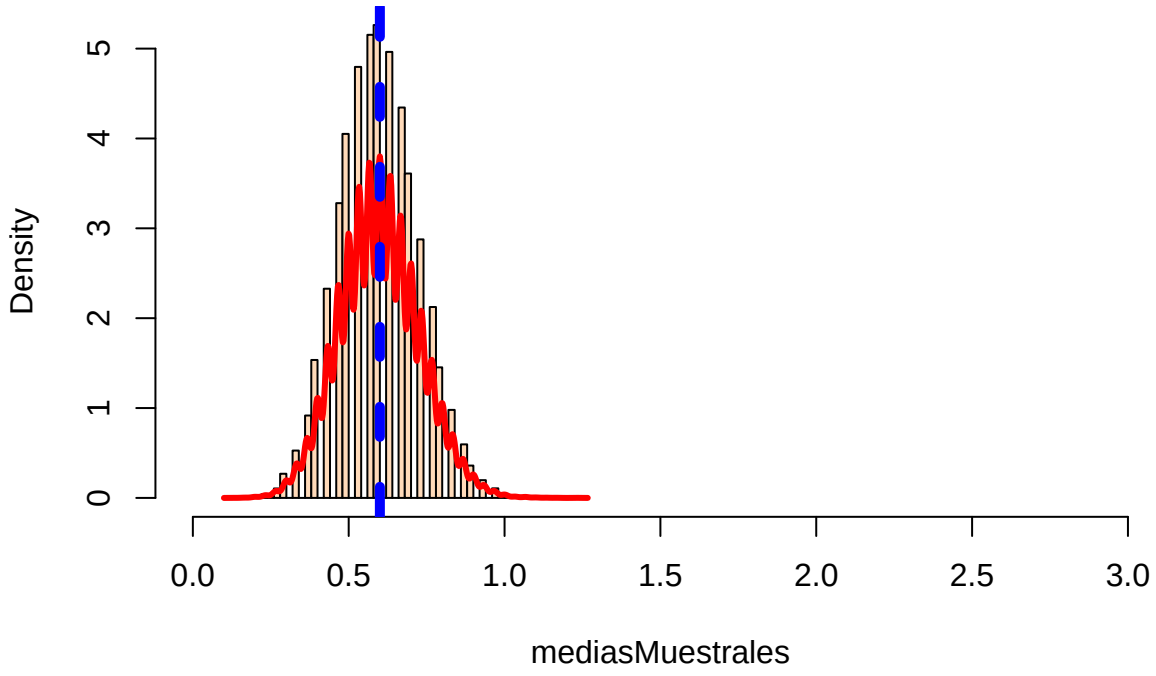
Observamos las medias de nuestra muestra de tamaño 30 en las diez primeras iteraciones del replicate:

```
head(mediasMuestrales, 10)
```

```
##  [1] 0.5000000 0.5666667 0.7333333 0.7333333 0.7333333 0.5666667 0.5333333
##  [8] 0.7000000 0.6333333 0.3333333
```

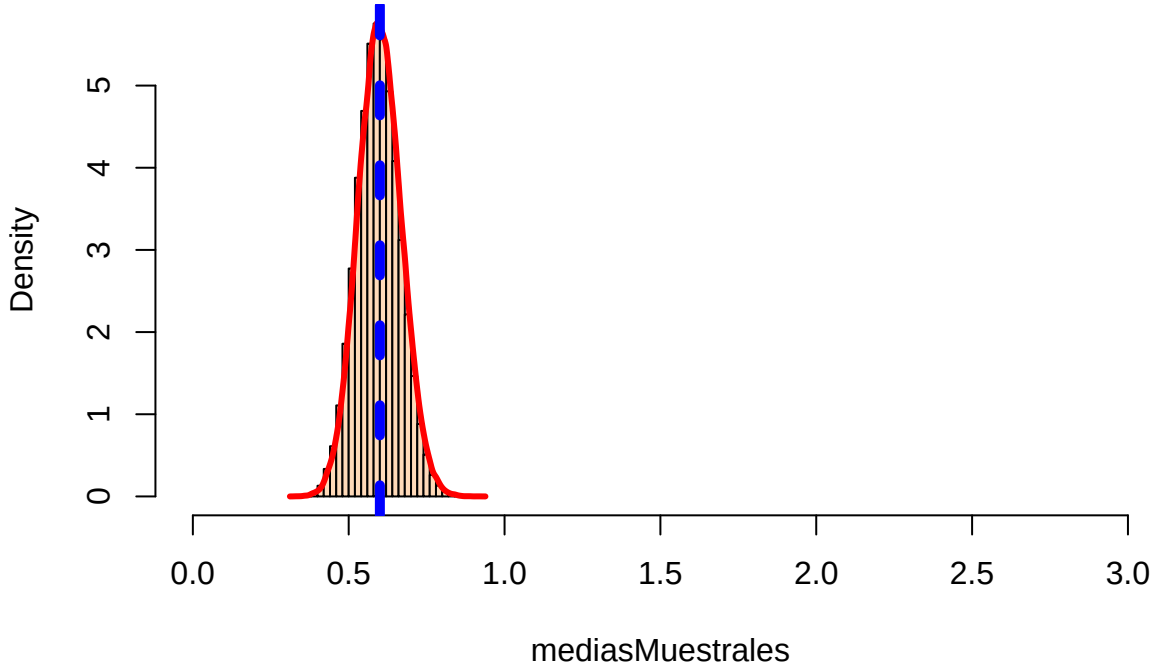
El histograma visualiza el resultado de las todas las medias muestrales obtenidas, sobre el que aparece pintada la línea azul discontinua que muestra la media teórica.

```
hist(mediasMuestrales, breaks = 40, main = "",
     col = "peachpuff", probability = TRUE, xlim = range(0:3))
lines(density(mediasMuestrales), lwd = 3, col = "red")
abline(v = media, lty=2, lwd=5, col="blue")
```



Repetimos el experimento con muestras de tamaño 100 y tenemos el siguiente resultado:

```
hist(mediasMuestrales, breaks = 40, main = "",
     col = "peachpuff", probability = TRUE, xlim = range(0:3))
lines(density(mediasMuestrales), lwd = 3, col = "red")
abline(v = media, lty=2, lwd=5, col="blue")
```



Con muestras de mayor tamaño se aprecia mejor el objetivo de este problema, comprobar el Teorema Central del Límite. Esto es, la media de las medias muestrales coincide con la media de la población.y prácticamente no hay muestras malas. Es extremadamente improbable que unamuestra elegida al azar sea muy mala. Además se observa que la distribución de las medias muestrales tiene forma de campana (y es muy estrecha).

- **Apartado 3: La variable aleatoria discreta X2 tiene esta tabla de densidad de probabilidad. Suponemos que X1 y X2 son independientes. ¿Qué valores puede tomar la suma X1 + X2? ¿Cuál es su tabla de probabilidad?**

valor de $X2$	0	1	2
Probabilidad de ese valor $P(X = x_i)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$

El rango valores que puede tomar la suma de ambos es de 0 a 5.

```r
x1 = c(0,1,2,3)
p1 = c(64/125, 48/125, 12/125, 1/125)
x2 = c(0,1,2)
p2 = c(1/2,1/4,1/4)
```

```r
(t1 = p2 %*% t(p1)) # Tabla con la probabilidad combinada de los distintos sucesos
```

```
##       [,1]  [,2]  [,3]  [,4]
## [1,] 0.256 0.192 0.048 0.004
## [2,] 0.128 0.096 0.024 0.002
## [3,] 0.128 0.096 0.024 0.002
```

```r
(t2 = outer(x2, x1, FUN = "+")) # Tabla con la suma de los números de x1 y x2
```

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

```r
(datos <- data.frame # Asociamos las diferentes sumas con su probabilidad
         (probab = c(t1), suma = c(t2)))
```

```
##    probab suma
## 1   0.256    0
## 2   0.128    1
## 3   0.128    2
## 4   0.192    1
## 5   0.096    2
## 6   0.096    3
## 7   0.048    2
## 8   0.024    3
## 9   0.024    4
## 10  0.004    3
## 11  0.002    4
## 12  0.002    5
```

```r
(probabilidades <- datos %>% # Agrupamos la tabla anterior y sumamos la probabilidad
                   group_by(suma = suma) %>%
                   summarize(prob = sum(probab)))
```

```
## # A tibble: 6 x 2
##    suma  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 X1 + X2. 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.**

La media teórica de la suma coincide con la suma de las medias, por lo tanto:

```
media1 = as.numeric(x1 %*% p1)
media2 = as.numeric(x2 %*% p2)
(mediaSuma = media1 + media2)
```

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

Lo comprobamos también con el producto escalar del resultado obtenido en el apartado anterior y vemos que, efectivamente coinciden.

```
(mediaSuma2 = as.numeric(probabilidades$suma %*% probabilidades$prob))
```

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

Utilizamos los métodos de sample y replicate para simular cien mil valores de esta variable suma y hacemos la media de las mediasMuestrales:

```
k = 100000

mediasMuestrales = replicate(k, {
  muestra1 = sample(x1, size = 1, replace = TRUE, prob = p1)
  muestra2 = sample(x2, size = 1, replace = TRUE, prob = p2)
  muestra1 + muestra2
})

mean(mediasMuestrales)
```

```
## [1] 1.34801
```

El resultado es muy similar al obtenido de manera teórica, validando así nuestro modelo.

Ejercicio 2. Datos limpios.

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.

En primer lugar, mostramos las 6 primeras observaciones de la tabla para poder comprobar la presencia de datos no limpios en la misma.

```r
datos <- read_csv("data/testResults.csv")
head(datos, 6)
```

```
## # 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
```

Como se ha podido comprobar, la columna "gender_age" presenta información de dos variables distintas: "gender" y "age", por lo que las vamos a separar. Por otro lado, en vez de tener 5 variables (columnas) para cada una de las semanas, haremos una tabla más larga que contenga información para cada semana y su correspondiente valor.

La implementación del código R que lleva a cabo dichas modificaciones es el siguiente:

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

```r
head(datosLimpios, 15)
```

```
## # A tibble: 15 x 7
##    name       id gender   age test_number week_number value
##    <chr>   <dbl> <chr>  <int>       <dbl> <chr>       <dbl>
##  1 Jacob     108 m         20           1 week1           8
##  2 Jacob     108 m         20           1 week2           5
##  3 Jacob     108 m         20           1 week3           7
##  4 Jacob     108 m         20           1 week4           5
##  5 Jacob     108 m         20           1 week5           6
##  6 Jacob     108 m         20           2 week1           2
##  7 Jacob     108 m         20           2 week2           2
##  8 Jacob     108 m         20           2 week3           4
##  9 Jacob     108 m         20           2 week4           0
## 10 Jacob     108 m         20           2 week5           3
## 11 Michael   490 m         19           1 week1          10
## 12 Michael   490 m         19           1 week2           0
## 13 Michael   490 m         19           1 week3           5
## 14 Michael   490 m         19           1 week4           4
## 15 Michael   490 m         19           1 week5           0
```

Se comprueba una tabla más larga, pero con los datos mucho más limpios para trabajar con ellos y llevar a cabo cualquier tipo de análisis.

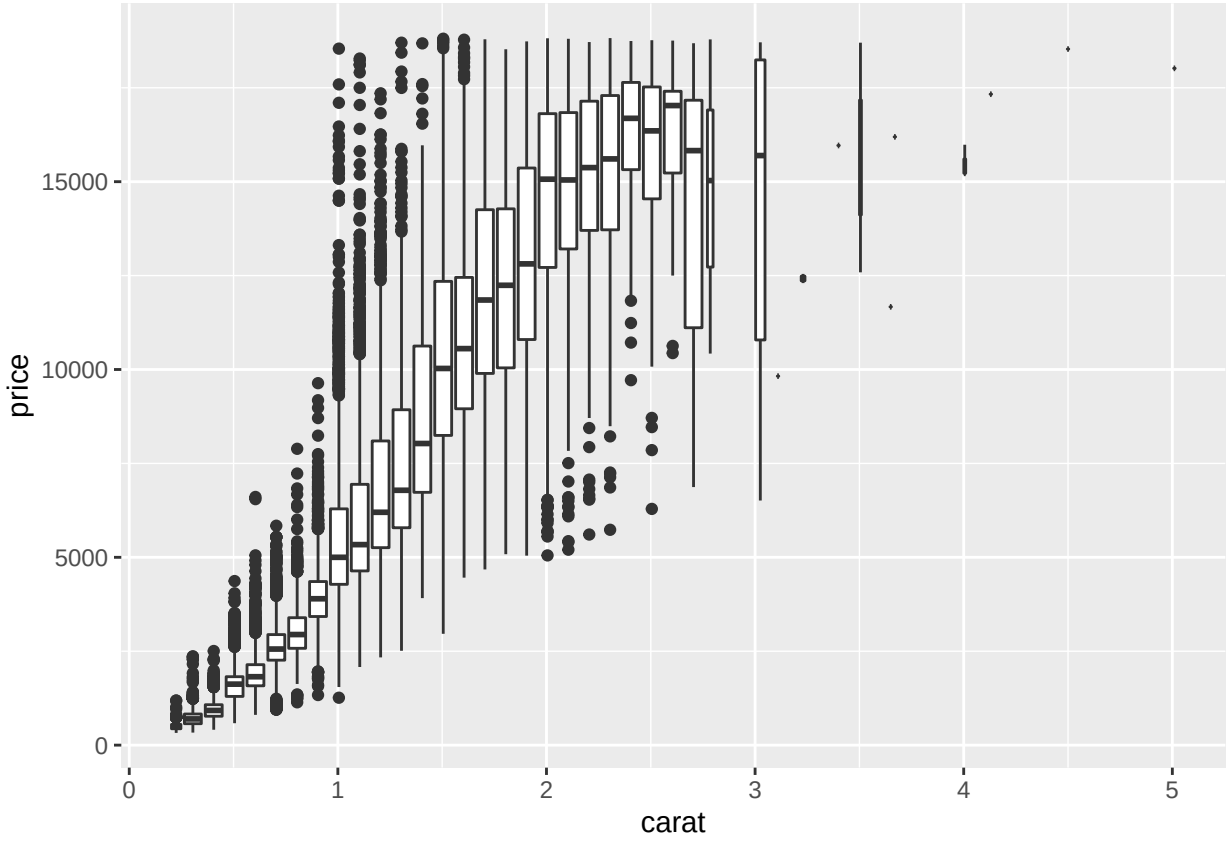
Ejercicio 3. Lectura de R4DS.

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.**

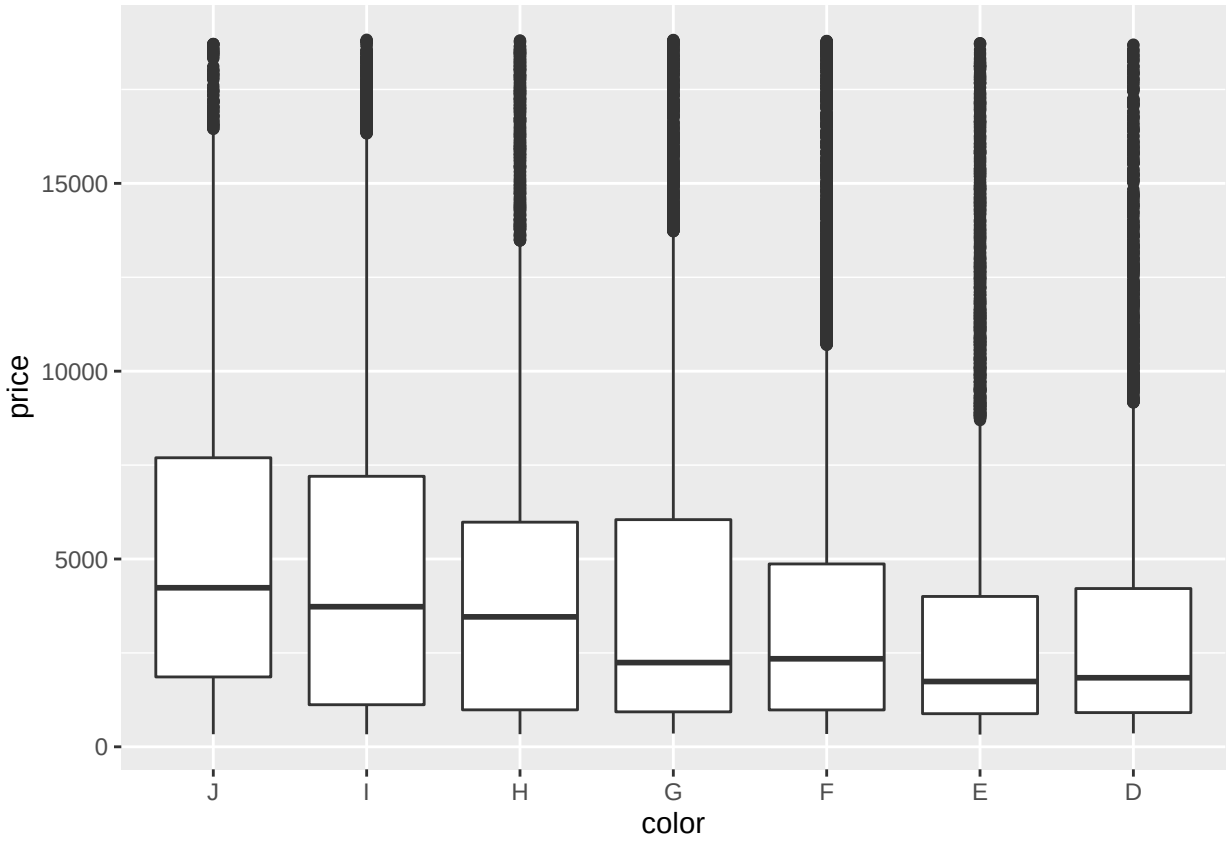
El capítulo sugiere la visualización de las distintas variables continuas de la siguiente manera:

```
ggplot(data = diamonds, mapping = aes(x = carat, y = price)) +
  geom_boxplot(mapping = aes(group = cut_width(carat, 0.1)), orientation = "x")
```



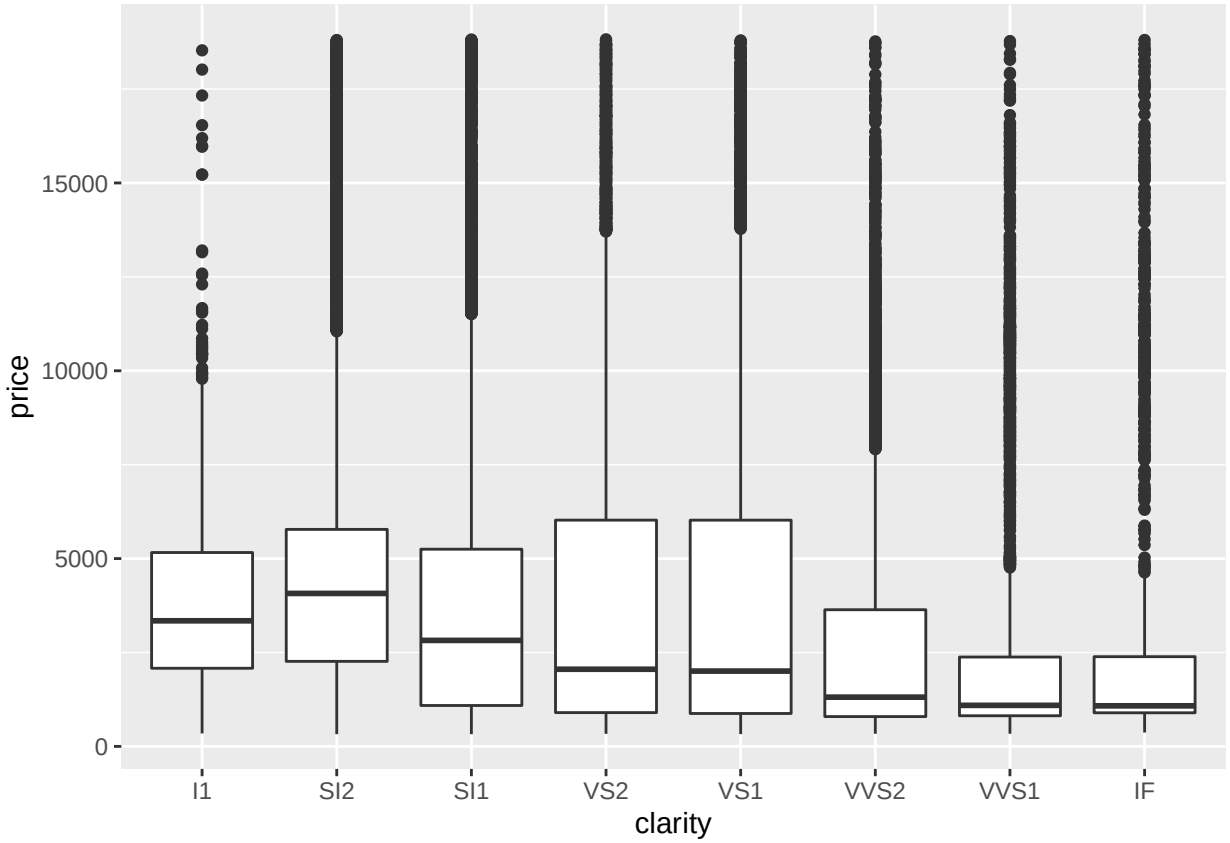
Observamos una relación negativa y débil entre la variable color y precio. La escala va de peor (J) a mejor (D). Por tanto, los niveles de color están en el orden equivocado. Para ello, se revierten el orden del nivel de color a lo largo del eje X.

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



También observamos una relación negativa débil entre la variable claridad y precio. La escala de claridad va de mejor (IF) a peor (I1).

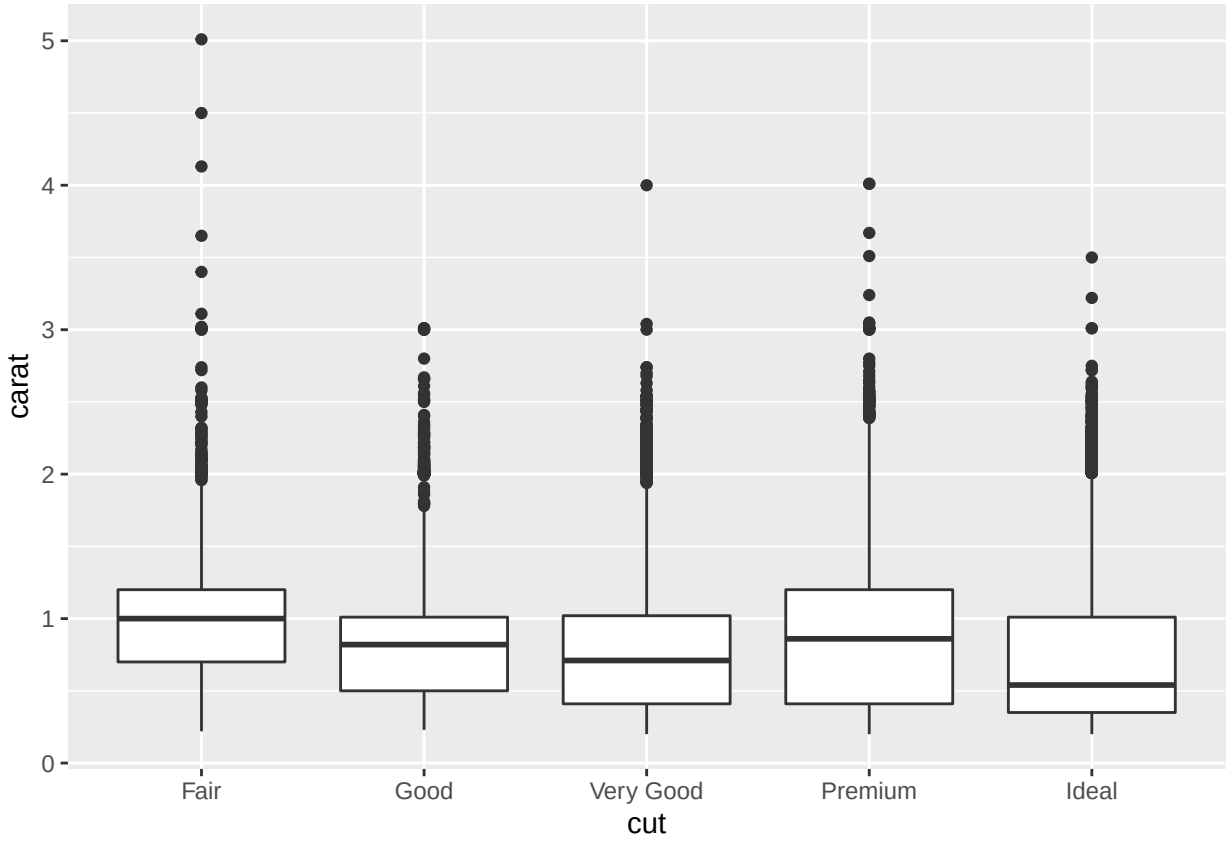
```
ggplot(data = diamonds) +
  geom_boxplot(mapping = aes(x = clarity, y = price))
```



Tanto para la variable claridad como para la variable color, hay mucha más variación entre cada categoría que entre todas las categorías. Por tanto, "carat" es el mejor predictor de los precios de diamante.

Ahora vemos la correlación de esta variable con "cut":

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



Se comprueba una ligera relación negativa entre ambas variables. Notablemente, los diamantes con la variable "carat" más alta, tienen el menor "cut" (Fair).

Podemos extraer como conclusión que un diamante más pequeño, requiere de un mejor corte, mientras que uno más grande puede ser vendido con una calidad menor.

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

En primer lugar, utilizamos el código empleado en la sección del libro para trabajar con datos limpios:

```
whov1 <- who %>%
  pivot_longer(
    cols = new_sp_m014:newrel_f65,
    names_to = "key",
    values_to = "number_of_cases",
    values_drop_na = TRUE
  )
```

```
whov2 <- whov1 %>%
  mutate(names_from = stringr::str_replace(key, "newrel", "new_rel"))
```

```
whov3 <- whov2 %>%
  separate(key, c("new", "type", "sexage"), sep = "_")
```

```
whov4 <- whov3 %>%
  select(-new, -iso2, -iso3)
```

```
whov5 <- whov4 %>%
  separate(sexage, c("sex", "age"), sep = 1)
```

Para cada país, año y sexo calculamos el numero total de casos. Puesto que la información de años anteriores al 1995 representan un 0.55% respecto al número total de observaciones (420 frente a 76046) filtraremos por esta condición para una mejor visualización de los datos.

```
whov5 %>%
  group_by(country, year, sex) %>%
  filter(year > 1994) %>%
  summarise(cases = sum(number_of_cases)) %>%
  unite(country_and_sex, country, sex, remove = FALSE) %>%
  ggplot(aes(x = year, y = cases, group = country_and_sex, color = sex)) +
  geom_line()
```