Tarea2

Gonzalo Ruiz Espinar

9/21/2021

Ejercicio 1

Apartado 1

Empezaremos cargamos la libreria *tidyverse*.

```
library(tidyverse)
```

Creamos un vector llamado `valor` y otro de probabilidad llamado `prob`:

```
valor = c(0,1,2,3)
prob = c(64/125, 48/125, 12/125, 1/125)
```

La media se calcula con el producto escalar:

```
media = sum(valor * prob)
media
```

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

La varianza teórica se puede calcular de la siguiente forma:

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

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

Apartado 2

Creamos un vector llamado `valor` y otro de probabilidad llamado `prob`:

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

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

Combinamos `sample` con `replicate` para simular cien mil muestras,`k`, de tamaño 10 o 30, `n` de esta variable X1.

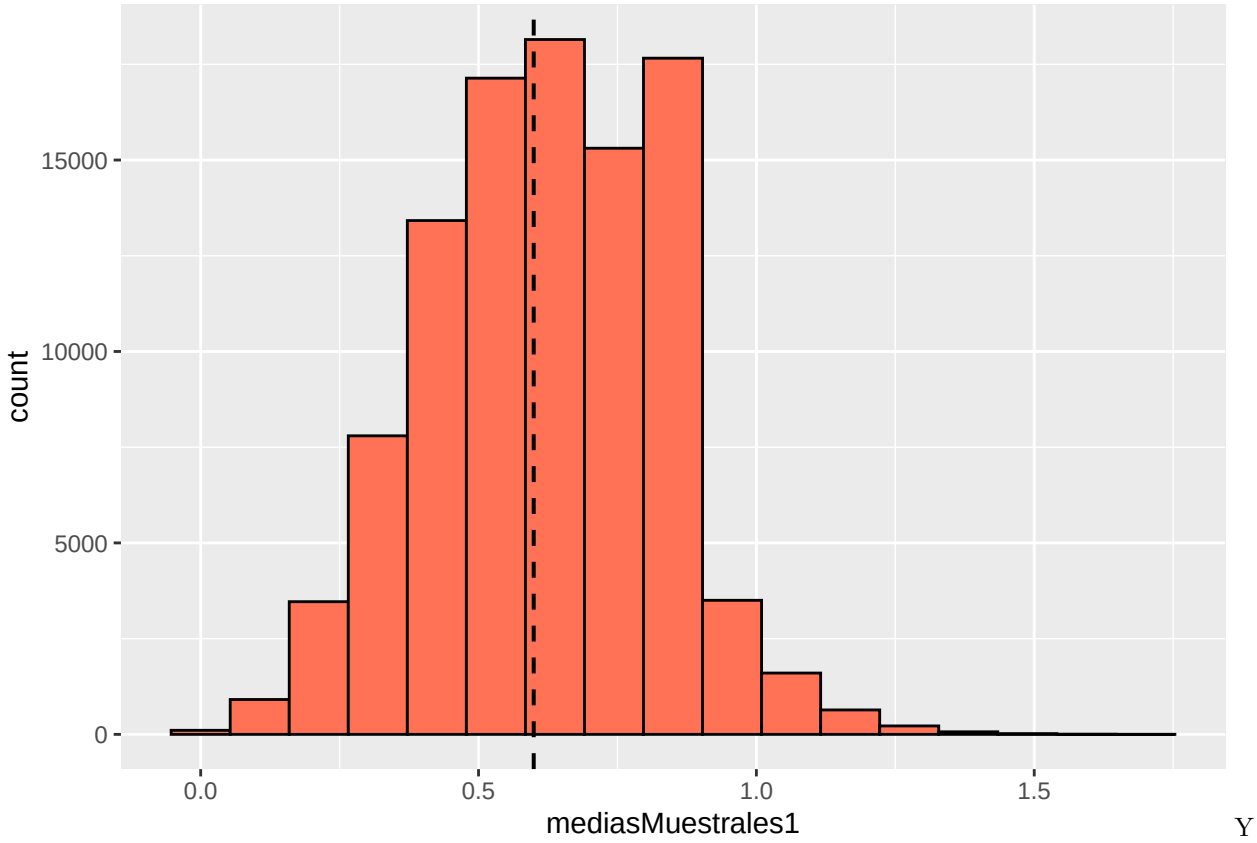
```
k = 100000
n1 = 10
n2 = 30
mediasMuestrales1 = replicate (k, {
  muestra = sample(valor, n1, replace = TRUE,prob = c(64/125, 48/125, 12/125, 1/125))
  mean(muestra)
})

mediasMuestrales2 = replicate (k, {
  muestra = sample(valor, n2, replace = TRUE,prob = c(64/125, 48/125, 12/125, 1/125))
```

```
  mean(muestra)
})
```

Estudiamos la distribución de las medias muestrales para muestras de 10

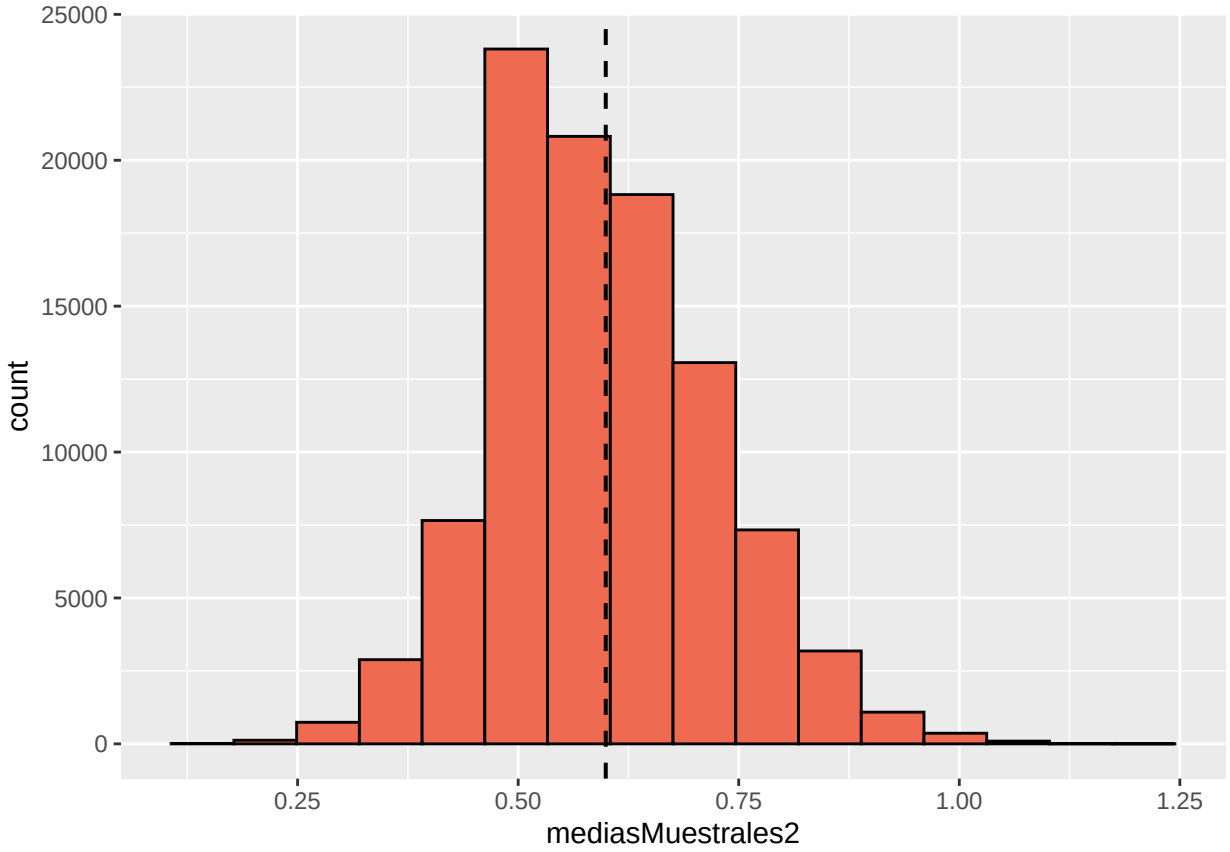
```
ggplot() +
  geom_histogram(aes(x = mediasMuestrales1), bins = 17, fill="coral1", color="black") +
  geom_vline(xintercept = mean(mediasMuestrales1), col="black", linetype="dashed", size=0.7)
```



Y

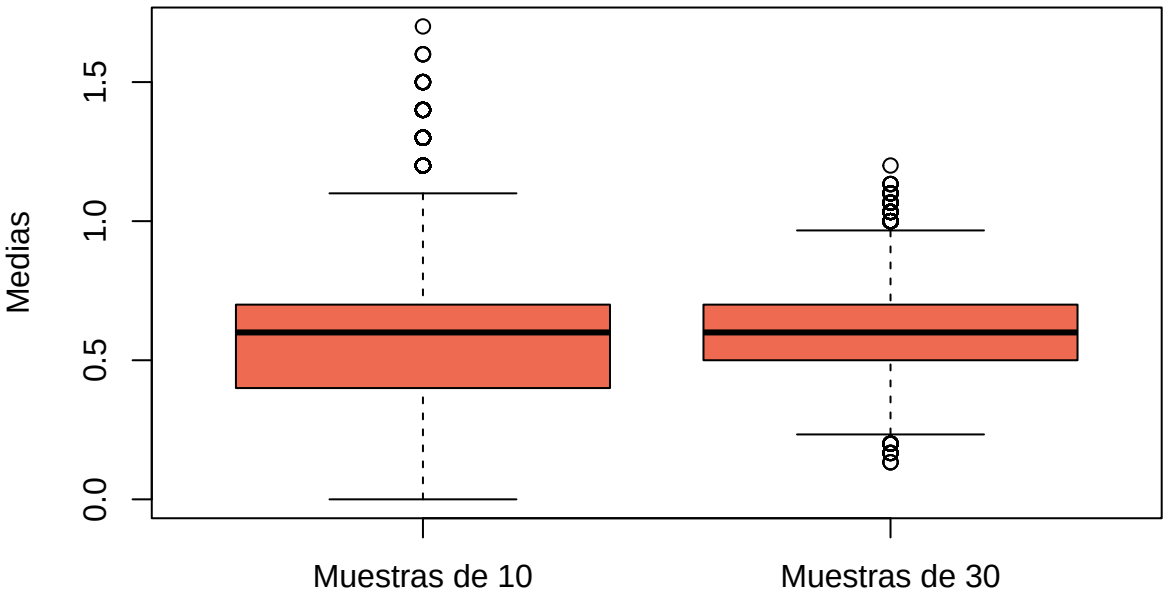
para medias muestrales para muestras de 30 repetimos el histograma.

```
ggplot() +
  geom_histogram(aes(x = mediasMuestrales2), bins = 16, fill="coral2", color="black") +
  geom_vline(xintercept = mean(mediasMuestrales1), col="black", linetype="dashed", size=0.7)
```



Otra gráfica que nos ayuda a ilustrar los datos es el *boxplot*.

```
bxp_cty = boxplot(na.omit(mediasMuestrales1), mediasMuestrales2, col="coral2",names = c("Muestras de 10
```



Por ejemplo podemos ver como cuando aumentamos el tamaño de las muestras (de 10 a 30) la varianza de la distribución de la media muestral disminuye. De hecho, sabemos que la desviación estándar de la distribución de la media muestral cumple que $s_1/s_2 \sim \sqrt{n_2/n_1} = \sqrt{3}$, lo cual se cumple en nuestro caso con error menor de un 1%:

```
(sqrt(3) - sd(mediasMuestrales1)/sd(mediasMuestrales2)) / sqrt(3)
```

```
## [1] 0.001259761
```

Apartado 3

Suponiendo que X1 y X2, para saber los posibles valores de la suma usamos el mínimo valor de cada variable, 0, y el máximo de cada una, 2 y 3. Es decir que la suma puede ir de 0 a 5. Para obtener la tabla de probabilidades realizaremos el siguiente cálculo.

```
valor1 = c(0,1,2,3)
valor2 = c(0,1,2)
prob1 = c(64/125, 48/125, 12/125, 1/125)
prob2 = c(1/2,1/4,1/4)
```

En primer lugar realizamos una tabla con todos los posibles resultados que podemos obtener de cada variable sumado. Por ejemplo: en la posición 13 (empezando a contar desde el 0) de la matriz tenemos 1+3=4, en la posición 31 tenemos 3+1=4 o en la posición 21 tenemos 2+1=3.

```
posSum = outer(valor2,valor1, FUN = "+")
posSum
```

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

A continuación creamos otra tabla donde se incluye la multiplicación de todas las posibles probabilidades de obtener un resultado concreto en cada variable. Por ejemplo: en la posición 12 (empezando a contar desde el 0) de la matriz tenemos la probabilidad de que obtengamos en la primera variable un 1 y en la segunda un 2, $\frac{48}{125} \cdot \frac{1}{4} = \frac{18}{125}$.

```
posProb = prob2 %*% t(prob1)
posProb
```

```
##       [,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
```

Pasamos estas matrices a un *dataframe* donde tengamos 2 columnas, una con las posibles sumas y otra con las probabilidades de obtener un resultado concreto en cada variable.

```
data.frame(c(posProb),c(posSum)) -> data
```

Un vez que tenemos la tabla `data`, calculamos la tabla de probabilidades de la suma de X1 y X2, agrupando todas las posibles sumas, y sumando las probabilidades asociadas a cada suma.

```
data %>% group_by(dado=c.posSum.) %>% summarize(prob=sum(c.posProb.))-> probabilidades
probabilidades
```

```
## # A tibble: 6 x 2
##    dado  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

Una vez que tenemos la tabla de probabilidades, calculamos la media de la suma de X1 y X1 de la forma habitual:

```
weighted.mean(probabilidades$dado,probabilidades$prob )
```

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

Para comprobar que hemos realizado bien el análisis, vamos a simular este caso. Generamos dos conjuntos X1 y X1 y los sumamos.

```
k = 100000
X1 = sample(valor1, k, prob = prob1, replace = TRUE)
X2 = sample(valor2, k, prob = prob2, replace = TRUE)
suma = X1+X2
mean(suma)
```

```
## [1] 1.34674
```

Vemos que obtenemos la misma media que el cálculo teórico.

Ejercicio 2

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.

```
library(readr)
```

Empezamos haciendo una pequeña visualización de los datos para hacernos una idea de la tabla:

```
datos <- 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(datos)
```

```
## # 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
```

En primer lugar separamos la columna `gender_age` en dos columna, `gender` y `age`, y a continuación transformamos las columnas de `week` en dos columnas que dices que semana es y el valor asociado.

```
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")
```

Después transformamos la columna de `week`y la de `gender` en factores.

```
datosLimpios$gender=factor( datosLimpios$gender, levels = c("m","f"),labels = c("Male","Female") )
datosLimpios$week_number=factor(datosLimpios$week_number,levels = paste("week", 1:7, sep = ""),labels =
                                  paste("week", 1:7, sep = "") )
```

Mostramos el resultado

```
head(datosLimpios)
```

```
## # A tibble: 6 x 7
##   name     id gender   age test_number week_number value
##   <chr> <dbl> <fct>  <int>       <dbl> <fct>       <dbl>
## 1 Jacob   108 Male      20           1 week1           8
## 2 Jacob   108 Male      20           1 week2           5
## 3 Jacob   108 Male      20           1 week3           7
## 4 Jacob   108 Male      20           1 week4           5
## 5 Jacob   108 Male      20           1 week5           6
## 6 Jacob   108 Male      20           2 week1           2
```

Ejercicio 3

Apartado 1

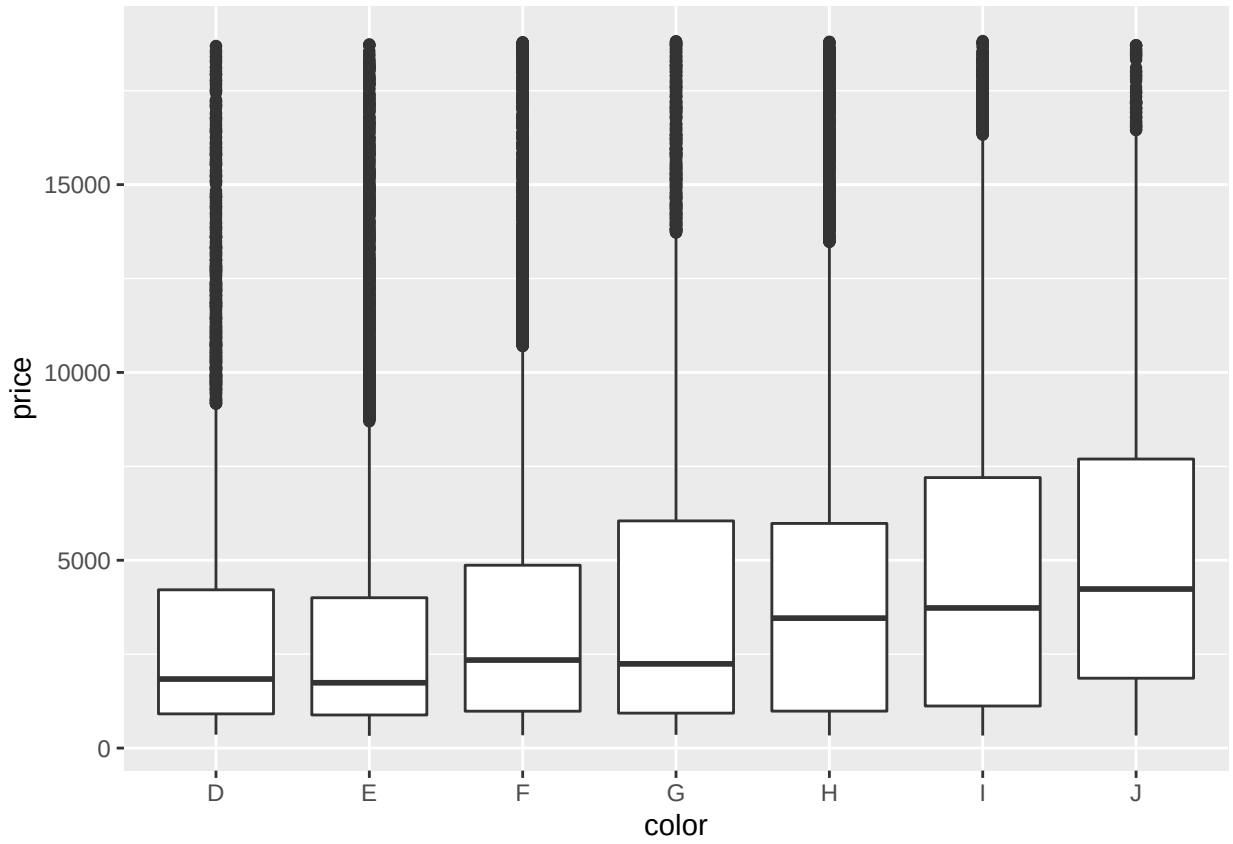
Nuestro objetivo es determinar que variable influye más en el precio. Empezaremos una limpieza de los datos, `diamonds`:

```
head(diamonds)
```

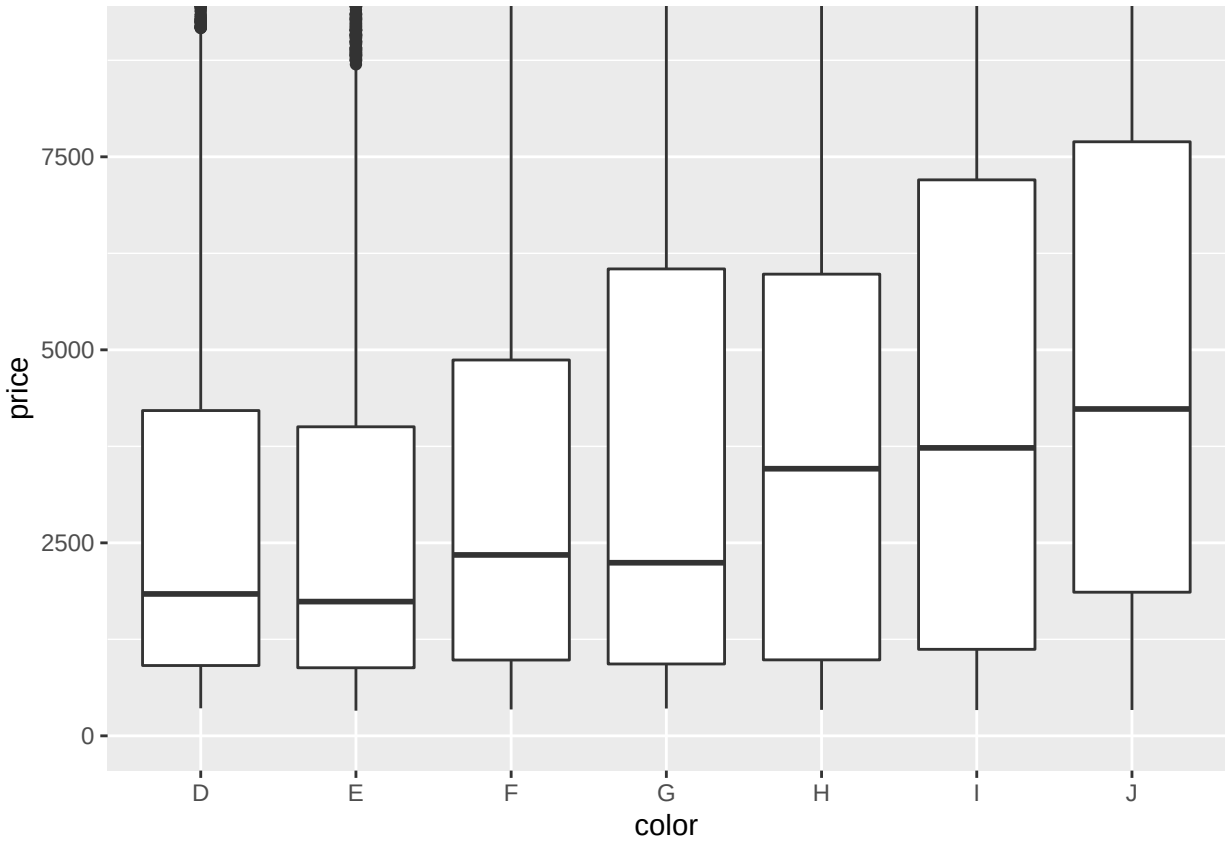
```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  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
```

Ya que la profundidad `depth`, la altura `table`y las longitudes `x`, `y` y `z` dependen de alguna manera del `carat`, solo consideraremos esta variables relativa a las dimensiones. Solo consideraremos pues `carat`, `clarity`, `color` y `cut`. En primer lugar compararemos el precio y el color de los diamantes. Vemos como no hay una gran variación de la mediana de los valores del precio y además tienen un primer y tercer cuartil relativamente grande lo cual nos dice que no podemos concluir que la variación del precio es grande.

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

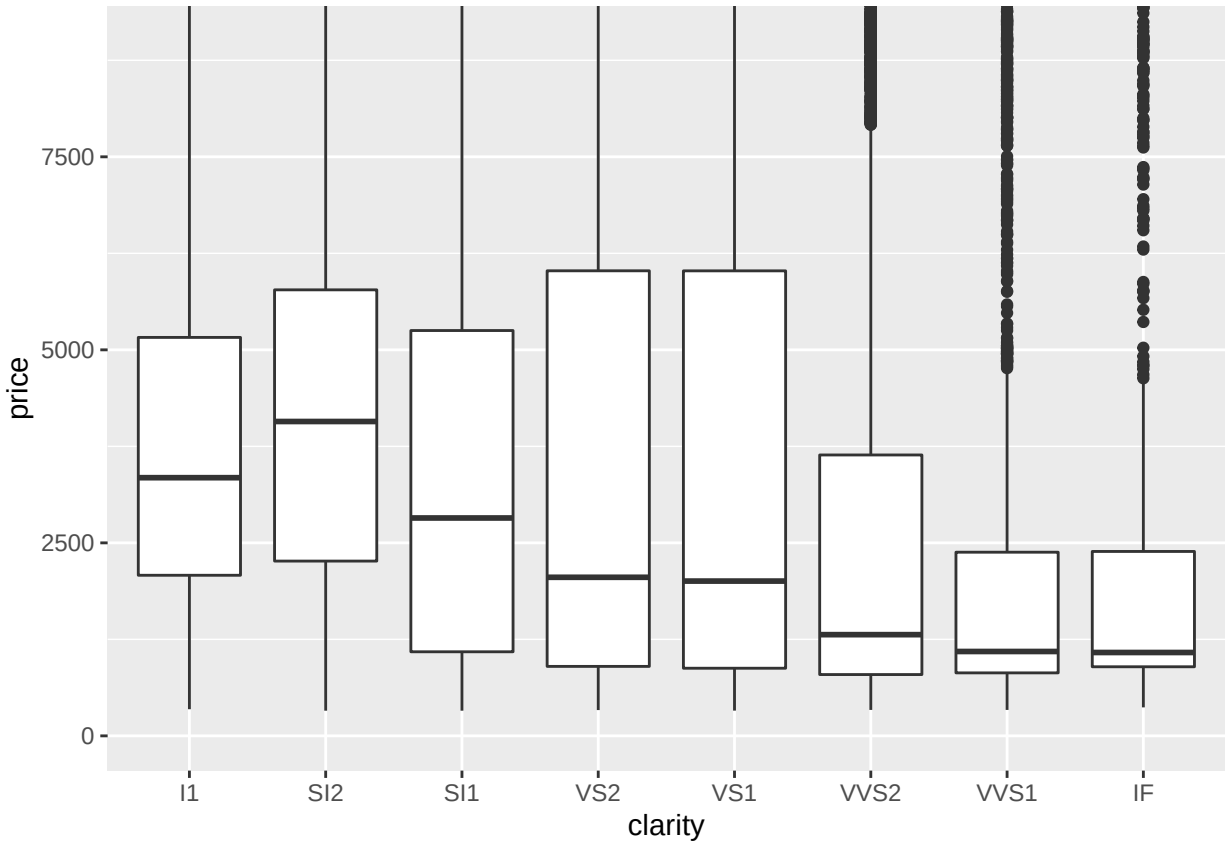



```
ggplot(diamonds)+
  geom_boxplot(mapping = aes(x = color, y = price))+
  coord_cartesian(ylim = c(0, 9000))
```



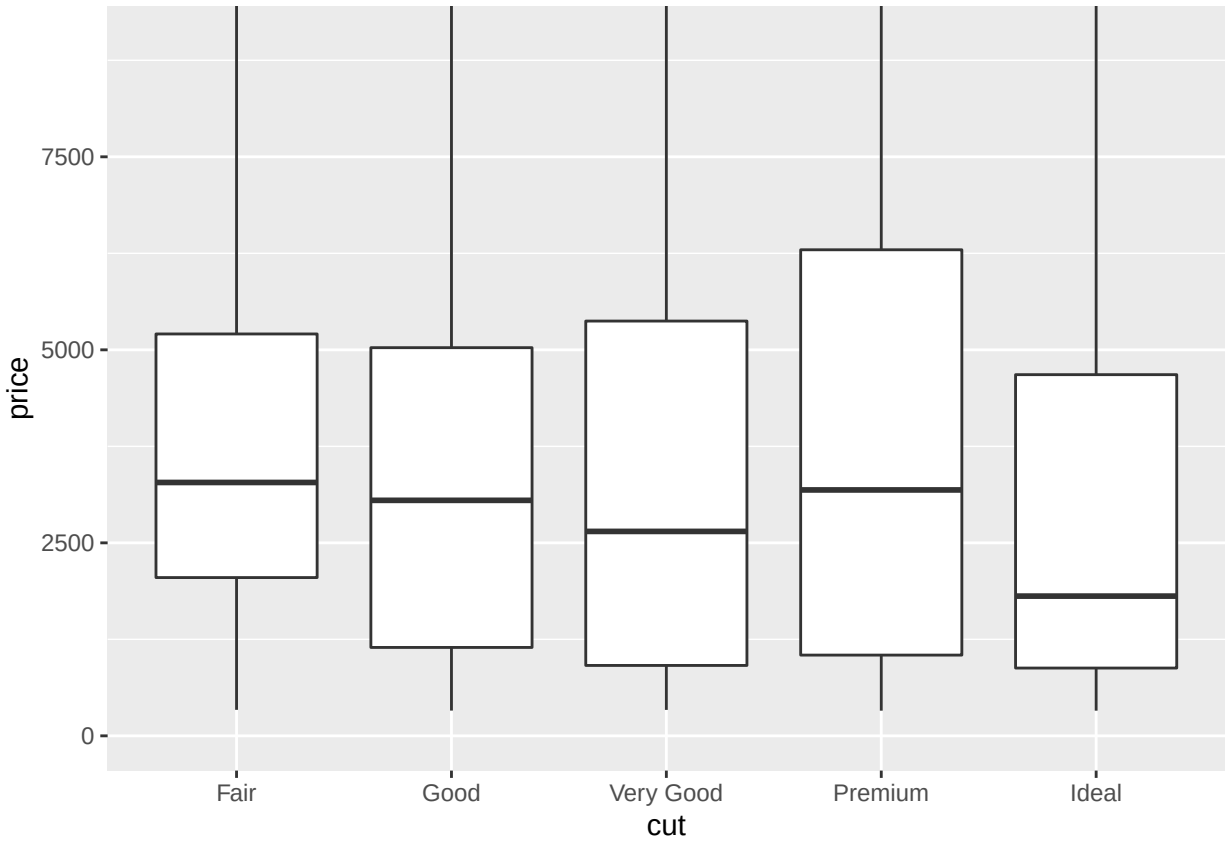
Por otro lado tenemos la claridad, donde sucede algo parecido al caso anterior y por tanto, aunque aumente el precio con la claridad, la variación no es muy grande.

```
ggplot(diamonds)+
  geom_boxplot(mapping = aes(x = clarity, y = price))+
  coord_cartesian(ylim = c(0, 9000))
```



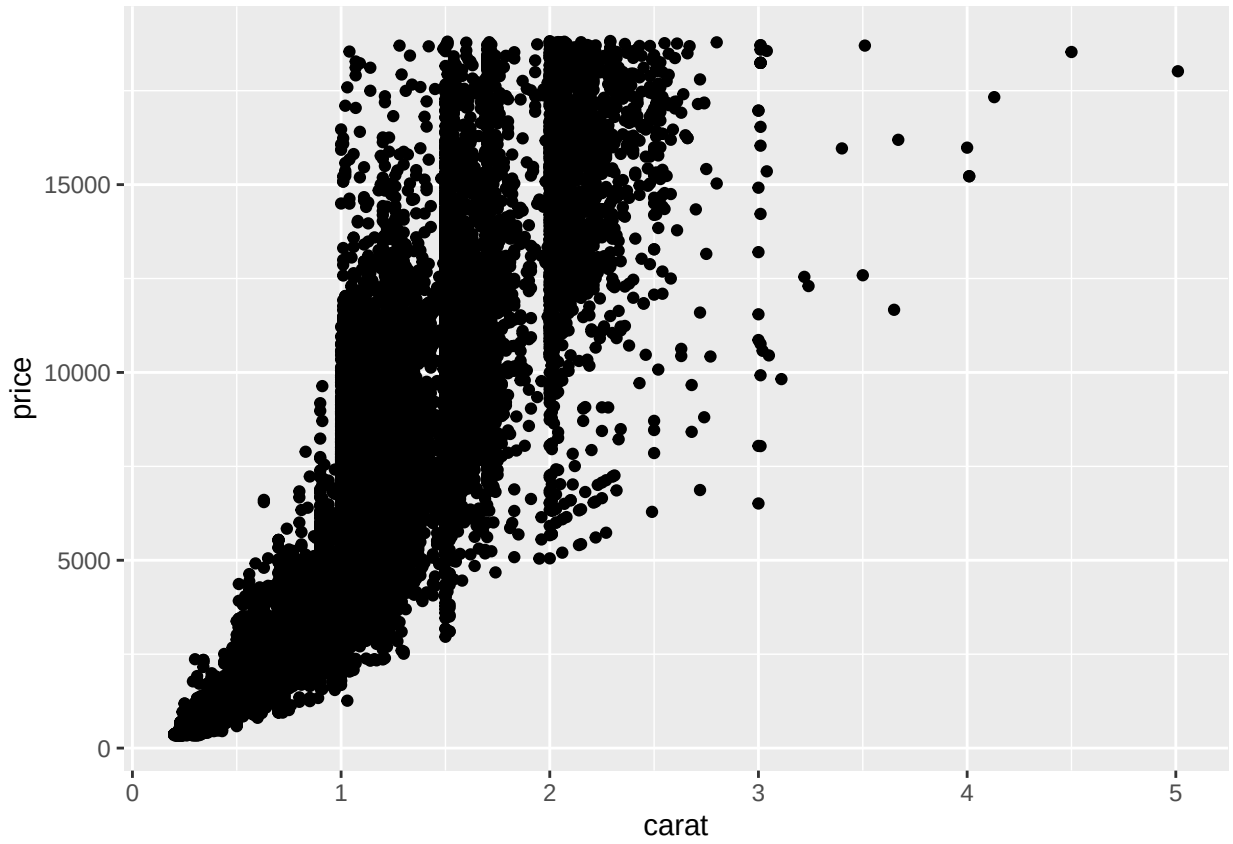
Por otro lado tenemos el corte similar al coso anterior.

```
ggplot(diamonds)+
  geom_boxplot(mapping = aes(x = cut, y = price))+
  coord_cartesian(ylim = c(0, 9000))
```



Por último tenemos el peso, que como podemos ver en la gráfica, tiene una tendencia lineal y por tanto podemos concluir que es la variable que mas influye en el precio.

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



Apartado 2

Tenemos que visualizar la en número de casos de tuberculosis por país, año y sexo. Empezaremos una limpieza de los datos, `who` y `population`:

```
head(population)
```

```
## # A tibble: 6 x 3
##   country      year population
##   <chr>       <int>      <int>
## 1 Afghanistan  1995   17586073
## 2 Afghanistan  1996   18415307
## 3 Afghanistan  1997   19021226
## 4 Afghanistan  1998   19496836
## 5 Afghanistan  1999   19987071
## 6 Afghanistan  2000   20595360
```

```
head(who)
```

```
## # 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>, ...
```

En primer lugar juntamos las columnas de los casos de tuberculosis en una nueva columna `TBcases`.

```
who %>%
  pivot_longer(starts_with("new"), names_to = "TBcases",values_drop_na = TRUE)-> whoLimpio
head(whoLimpio)
```

```
## # A tibble: 6 x 6
##   country     iso2  iso3   year TBcases      value
##   <chr>       <chr> <chr> <int> <chr>        <int>
## 1 Afghanistan AF    AFG    1997 new_sp_m014      0
## 2 Afghanistan AF    AFG    1997 new_sp_m1524    10
## 3 Afghanistan AF    AFG    1997 new_sp_m2534     6
## 4 Afghanistan AF    AFG    1997 new_sp_m3544     3
## 5 Afghanistan AF    AFG    1997 new_sp_m4554     5
## 6 Afghanistan AF    AFG    1997 new_sp_m5564     2
```

Como vemos, la columna de `TBcases` contiene información del género, rango de edad y tipo de tuberculosis. Procedemos a separarla:

```
 whoLimpio %>%
separate(TBcases, into = c("A", "MethodDiagnosis","GenderAux"), sep = c("_"), convert = TRUE) %>%
  separate(GenderAux,into = c("Gender", "AgeGroup"), sep = 1, convert = TRUE) %>%
select(-A) -> whoLimpio
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 2580 rows [243,
## 244, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 903,
## 904, 905, 906, ...].
```

```
head(whoLimpio)
```

```
## # A tibble: 6 x 8
##   country     iso2  iso3   year MethodDiagnosis Gender AgeGroup value
##   <chr>       <chr> <chr> <int> <chr>           <chr>     <int> <int>
## 1 Afghanistan AF    AFG    1997 sp              m            14     0
## 2 Afghanistan AF    AFG    1997 sp              m          1524    10
## 3 Afghanistan AF    AFG    1997 sp              m          2534     6
## 4 Afghanistan AF    AFG    1997 sp              m          3544     3
## 5 Afghanistan AF    AFG    1997 sp              m          4554     5
## 6 Afghanistan AF    AFG    1997 sp              m          5564     2
```

Convertimos a factor las columnas de `Gender`, `AgeGroup` y `MethodDiagnosis`:

```
whoLimpio$Gender=factor( whoLimpio$Gender, levels = c("m","f"),labels = c("Male","Female") )
whoLimpio$AgeGroup=factor( whoLimpio$AgeGroup, levels = c(14,1524,2534,3544,4554,5564,65),
                           labels = c("0-14","15-24","25-34","35-44","45-54","55-64","65") )
head(whoLimpio)
```

```
## # A tibble: 6 x 8
##   country     iso2  iso3   year MethodDiagnosis Gender AgeGroup value
##   <chr>       <chr> <chr> <int> <chr>           <fct>  <fct>    <int>
```

```
## 1 Afghanistan AF    AFG    1997 sp              Male   0-14         0
## 2 Afghanistan AF    AFG    1997 sp              Male   15-24       10
## 3 Afghanistan AF    AFG    1997 sp              Male   25-34        6
## 4 Afghanistan AF    AFG    1997 sp              Male   35-44        3
## 5 Afghanistan AF    AFG    1997 sp              Male   45-54        5
## 6 Afghanistan AF    AFG    1997 sp              Male   55-64        2
```

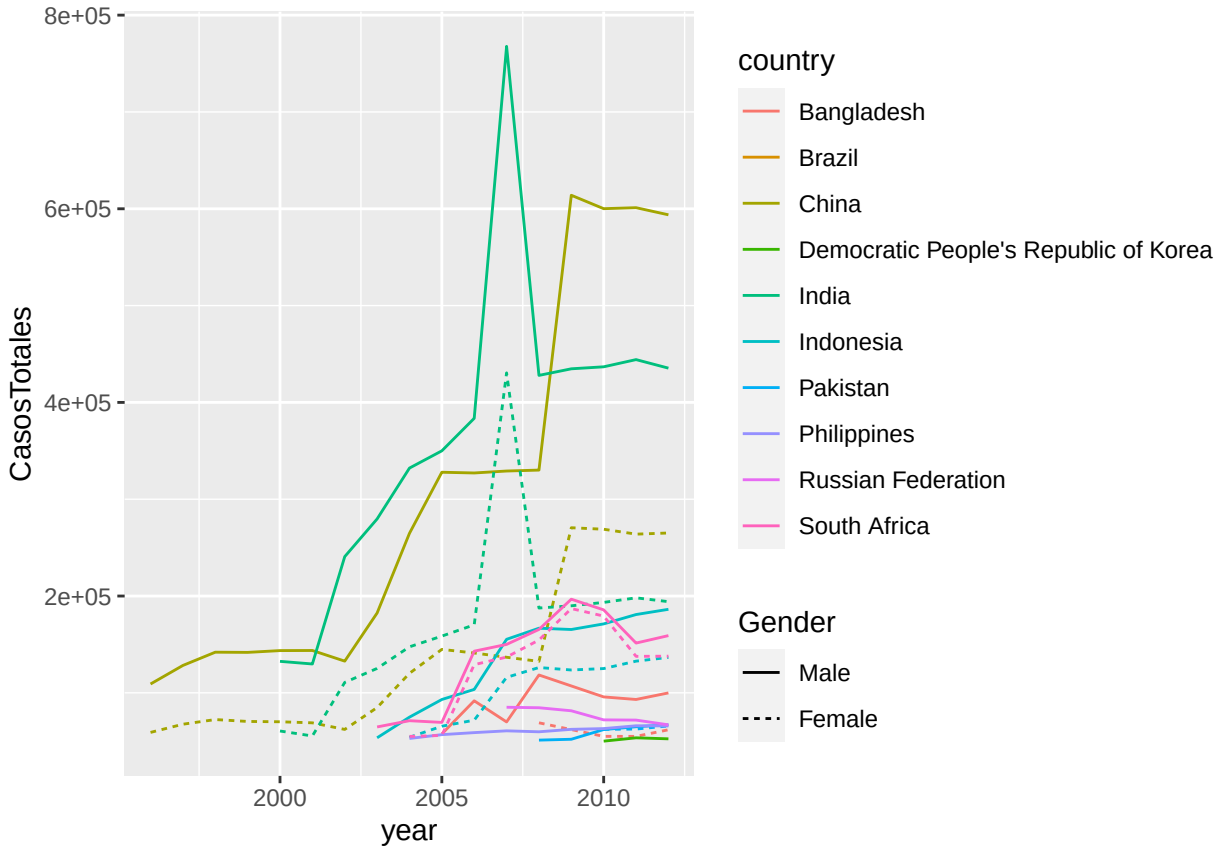
Dado que posteriormente estamos interesados en el número de casos por 100k habitantes, unimos las tablas `who` y `population`

```
whoLimpio<-left_join(whoLimpio,population, by=c("country"="country","year"="year"))
head(whoLimpio)
```

```
## # A tibble: 6 x 9
##   country     iso2  iso3   year MethodDiagnosis Gender AgeGroup value population
##   <chr>       <chr> <chr> <int> <chr>           <fct>  <fct>    <int>      <int>
## 1 Afghanistan AF    AFG    1997 sp              Male   0-14         0   19021226
## 2 Afghanistan AF    AFG    1997 sp              Male   15-24       10   19021226
## 3 Afghanistan AF    AFG    1997 sp              Male   25-34        6   19021226
## 4 Afghanistan AF    AFG    1997 sp              Male   35-44        3   19021226
## 5 Afghanistan AF    AFG    1997 sp              Male   45-54        5   19021226
## 6 Afghanistan AF    AFG    1997 sp              Male   55-64        2   19021226
```

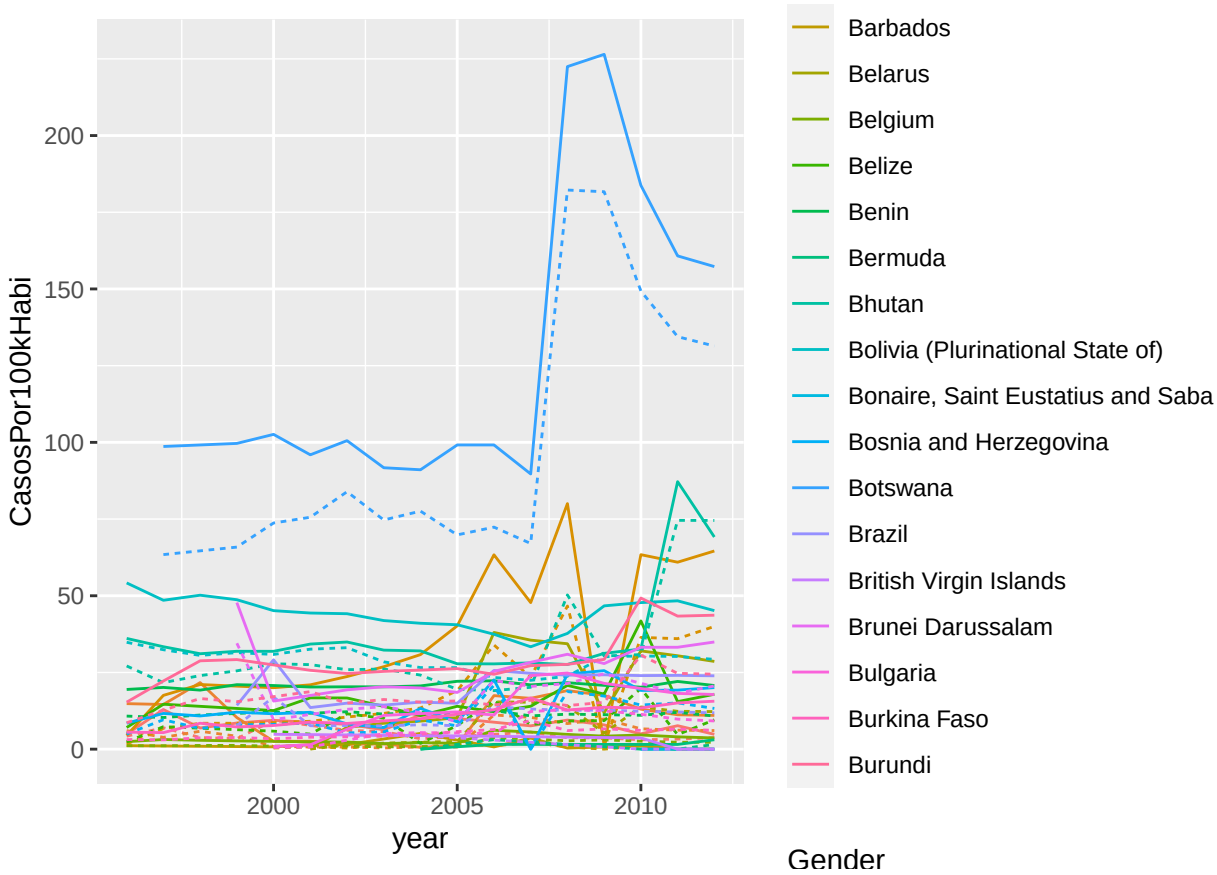
Dado que si graficamos a todos los paises, habría tanta información en la gráfica que no se podría obtener ninguna conclusión visual, hemos dedicidido filtrar de alguna manera nuestros datos. Por ejemplo, en primer lugar representamos la incidencia de tuberculosis de aquellos paises que tengan más de 50000 casos por año:

```
na.omit(whoLimpio)  %>% group_by(country,year,Gender) %>% summarise(CasosTotales=sum(value),CasosPor100
  filter(year>1995)  %>%  filter(CasosTotales>50000) %>%
 # filter(substr(country,1,2) == "Ba") %>%
ggplot(aes(x = year, y = CasosTotales)) +
  geom_line(aes(colour=country,linetype=Gender,  group=interaction(Gender, country)))
```



Aquí representamos la incidencia de tuberculosis de aquellos paises cuyo nombre empiece por la letra B:

```
na.omit(whoLimpio)  %>% group_by(country,year,Gender) %>%
  summarise(CasosTotales=sum(value),CasosPor100kHabi=sum(value)*100000/(population)) %>%
  filter(year>1995)   %>%
  filter(substr(country,1,1) == "B") %>%
  ggplot(aes(x = year, y = CasosPor100kHabi)) +
  geom_line(aes(colour=country,linetype=Gender,  group=interaction(Gender, country)))
```


CasosPor100kHabi
200
150
100
50
0
2000
2005
2010
year
Barbados
Belarus
Belgium
Belize
Benin
Bermuda
Bhutan
Bolivia (Plurinational State of)
Bonaire, Saint Eustatius and Saba
Bosnia and Herzegovina
Botswana
Brazil
British Virgin Islands
Brunei Darussalam
Bulgaria
Burkina Faso
Burundi
Gender