# Prueba de Evaluación Continua 3 (PEC3)

Francisco Javier Botey Bataller & Lucas Goiriz Beltrán

15/12/2020

# SOFTWARE PARA EL ANÁLISIS DE DATOS (SAD)

## MÁSTER UNIVERSITARIO EN BIOINFORMÁTICA Y BIOESTADÍSTICA

### Motivación y datasets empleados

El objetivo de nuestro trabajo es estudiar si existe alguna relación entre la vacuna BCG (*Bacillus de Calmette y Guérin*) para la tuberculosis y los datos de mortalidad de la COVID-19 en algunos países, ya que hay estudios que sugieren esta vacuna incrementa las capacidades inmunitarias de la población, hecho que se ve en el número reducido de fallecimientos por COVID-19 en ciertos países. Mediante los conjuntos de datos de *BCG* y de mortalidad por COVID-19 cedidos por *The BCG world atlas* y por BCG - COVID-19 AI Challenge de Kaggle, vamos a intentar desvelar dichas relaciones.

Los ficheros en cuestión son del tipo '`csv`, así que son fácilmente importables a data frames en`R`:

```
# Cargamos ambos datasets,

BCG_strain <-
  read_csv("task_2-BCG_strain_per_country-1Nov2020.csv")

COVID_noformat <-
  read_csv(
    "task_2-COVID-19-death_cases_per_country_after_fifth_death-till_22_September_2020.csv"
  )

# Intenté ver que hay dentro de los data frames, pero el print es feo así que lo
# escribiré a mano
# str(COVID_noformat)
# str(BCG_strain)
```

El contenido de las variables `BCG_strain` y `COVID_noformat` es el siguiente:

| BCG_strain | COVID_noformat |
|---|---|
| country_name | country_name |
| country_code | alpha_3_code |
| mandatory_bcg_strain_2015-2020 | date_first_death |
| mandatory_bcg_strain_2010-2015 | date_fifth_death |
| mandatory_bcg_strain_2005-2010 | deaths_per_million_10_days_after_fifth_death |
| mandatory_bcg_strain_2000-2005 | deaths_per_million_15_days_after_fifth_death |
| mandatory_bcg_strain_1990-2000 | deaths_per_million_20_days_after_fifth_death |
| mandatory_bcg_strain_1980-1990 | deaths_per_million_25_days_after_fifth_death |
| mandatory_bcg_strain_1970-1980 | deaths_per_million_30_days_after_fifth_death |
| mandatory_bcg_strain_1960-1970 | deaths_per_million_35_days_after_fifth_death |

| BCG_strain | COVID_noformat |
|---|---|
| mandatory_bcg_strain_1950-1960 | deaths_per_million_40_days_after_fifth_death |
| vaccination_timing_unified | deaths_per_million_45_days_after_fifth_death |
| BCG Atlas: Which year was vaccination introduced? | deaths_per_million_50_days_after_fifth_death |
| Year of changes to BCG schedule | deaths_per_million_55_days_after_fifth_death |
| BCG Atlas: BCG Recommendation Type | deaths_per_million_60_days_after_fifth_death |
| BCG Atlas: Details of changes | deaths_per_million_65_days_after_fifth_death |
| BCG Atlas: Timing of 1st BCG? | deaths_per_million_70_days_after_fifth_death |
| BCG Atlas: BCG Strain | deaths_per_million_75_days_after_fifth_death |
| BCG Atlas: How long has this BCG vaccine strain been used? | deaths_per_million_80_days_after_fifth_death |
| | deaths_per_million_85_days_after_fifth_death |
| | deaths_per_million_90_days_after_fifth_death |
| | deaths_per_million_95_days_after_fifth_death |
| | deaths_per_million_100_days_after_fifth_death |
| | deaths_per_million_105_days_after_fifth_death |
| | deaths_per_million_110_days_after_fifth_death |
| | deaths_per_million_115_days_after_fifth_death |
| | deaths_per_million_120_days_after_fifth_death |
| | deaths_per_million_125_days_after_fifth_death |
| | deaths_per_million_130_days_after_fifth_death |
| | deaths_per_million_135_days_after_fifth_death |
| | deaths_per_million_140_days_after_fifth_death |
| | deaths_per_million_145_days_after_fifth_death |
| | deaths_per_million_150_days_after_fifth_death |
| | stringency_index_10_days_after_fifth_death |
| | stringency_index_15_days_after_fifth_death |
| | stringency_index_20_days_after_fifth_death |
| | stringency_index_25_days_after_fifth_death |
| | stringency_index_30_days_after_fifth_death |
| | stringency_index_35_days_after_fifth_death |
| | stringency_index_40_days_after_fifth_death |
| | stringency_index_45_days_after_fifth_death |
| | stringency_index_50_days_after_fifth_death |
| | stringency_index_55_days_after_fifth_death |
| | stringency_index_60_days_after_fifth_death |
| | stringency_index_65_days_after_fifth_death |
| | stringency_index_70_days_after_fifth_death |
| | stringency_index_75_days_after_fifth_death |
| | stringency_index_80_days_after_fifth_death |
| | stringency_index_85_days_after_fifth_death |
| | stringency_index_90_days_after_fifth_death |
| | stringency_index_95_days_after_fifth_death |
| | stringency_index_100_days_after_fifth_death |
| | stringency_index_105_days_after_fifth_death |
| | stringency_index_110_days_after_fifth_death |
| | stringency_index_115_days_after_fifth_death |
| | stringency_index_120_days_after_fifth_death |
| | stringency_index_125_days_after_fifth_death |
| | stringency_index_130_days_after_fifth_death |
| | stringency_index_135_days_after_fifth_death |
| | stringency_index_140_days_after_fifth_death |
| | stringency_index_145_days_after_fifth_death |

| BCG_strain | COVID_noformat |
|---|---|
| | stringency_index_150_days_after_fifth_death |

Una visualización preliminar de estos datos revela que son todos del tipo `string` y que además muchas columnas sin datos (columnas cuyo único contenido es `NULL`), por lo tanto llevaremos a cabo una limpieza de los mismos además de cambios de tipo de variables para que las manipulaciones posteriores sean más cómodas. Los detalles se muestran en el siguiente bloque de código:

```
# Limpiar datos de BCG

# Elimino columnas que sean sólo NA
BCG_strain <- BCG_strain[, apply(!is.na(BCG_strain), 2, all)]

# De momento, no me interesa qué vacunas se ponían cada año, sino si se ponían o no.

# Transformo los valores de cada año en
# 0 - No se ponía vacuna, hasta ahora None
# 1 - Sí se ponía vacuna
# NA - Este dato es desconocido, hasta ahora Unknown

BCG_strain_no_strain <- BCG_strain
# Transformo los valores de las columnas
BCG_strain_no_strain[, -1] <-
    sapply(BCG_strain_no_strain[, -1], function(x) {
        a <-
            gsub("None", 0, x) %>% gsub("Unknown", NA, .) # Añado los 0 y los NA.
        for (i in 1:length(a)) {
            # Serán 1 aquellos que no sean ni 0 ni NA
            if (a[i] != "0" && !is.na(a[i])) {
                a[i] <- 1
            }
        }
        return(as.integer(a)) # Cambio las columnas a integer
    })
BCG_no_strain_no_NA <- na.omit(BCG_strain_no_strain)

# Versión más compacta del dataframe, sin datos a diferentes días o años.
# Agrupando los datos de vacunas en tres columnas:
# periods_with_vaccine - incluye el número de periodos estudiados con vacunación activa
# vaccination_2020_2015 - el único periodo con el que nos quedamos, el último
# first_vaccine_year - de los años estudiados, el primero con campaña de vacunación.
#     En el caso de no tener vacunación, este será el último año estudiado (2020)
# last_vaccine_year - de los años estudiados, el último con campaña de vacunación.
#     En el caso de no tener vacunación, este será el primer año estudiado (1950)

# Creamos el nuevo dataframe simplificado
BCG_no_strain_simple <- data.frame(
  "country_name" = BCG_no_strain_no_NA$country_name,
  "periods_with_vaccine" = BCG_no_strain_no_NA%>%
  .[2:ncol(.)] %>% rowSums(), # sumamos los periodos con vacuna
  "vaccination_2020_2015" = BCG_no_strain_no_NA$`mandatory_bcg_strain_2015-2020`)

# Añadimos el último año con vacunación
```

```r
BCG_no_strain_simple$last_vaccine_year <-
  (
  names(BCG_no_strain_no_NA[2:ncol(BCG_no_strain_no_NA)])
  [max.col(BCG_no_strain_no_NA[2:ncol(BCG_no_strain_no_NA)] != 0, 'first')] %>%
  substring(nchar(.) - 3, nchar(.)) %>%
  as.numeric()
)
# Añadimos el primer año con vacunación
BCG_no_strain_simple$first_vaccine_year <-
  (
  names(BCG_no_strain_no_NA[2:ncol(BCG_no_strain_no_NA)])
  [max.col(BCG_no_strain_no_NA[2:ncol(BCG_no_strain_no_NA)] != 0, 'last')] %>%
  substring(nchar(.) - 8, nchar(.) - 5) %>%
  as.numeric()
)

# El próximo código es necesario para que los países sin campaña de vacunación no
# obtengan los mejores valores. El código utilizado para obtener el último o primer
# año de vacunación les favorece, ya que obtiene el primer o el último índice de
# aquellos valores distintos de 0. Como en su caso no hay ningún valor distinto a 0,
# este sería simplemente el primero o el último. Para arreglar esto, establezco
# manualmente que tengan el último año de vacunación más bajo posible y el primer
# año de vacunación más alto posible.
BCG_no_strain_simple[BCG_no_strain_simple$periods_with_vaccine == 0,]$last_vaccine_year = 1950

BCG_no_strain_simple[BCG_no_strain_simple$periods_with_vaccine == 0,]$first_vaccine_year = 2020
################################################################################

# Limpiar datos de COVID

# Eliminamos columnas que sean sólo NA
COVID_noNA <- COVID_noformat[, apply(!is.na(COVID_noformat), 2, all)]

# En este caso, para variar, los valores vacíos están denotados como NULL,
# cambiamos esto a NA
COVID_Na <- sapply(COVID_noNA, function(x)
    gsub("NULL", NA, x))

# El resultado de la función anterior es una string. Lo convertimos a dataframe.
COVID_Na_df <- as.data.frame(COVID_Na)

# Modificamos las fechas para que se almacenen como Date
COVID_Na_df[, c("date_fifth_death")] <-
    as.Date(COVID_Na_df[, c("date_fifth_death")], "%d/%m/%y")

COVID_Na_df[, c("date_first_death")] <-
    as.Date(COVID_Na_df[, c("date_first_death")], "%d/%m/%y")

# Modificamos las muertes para que se almacenen como floats.
COVID_Na_df[, -c(1, 2, 3, 4)] <-
    sapply(COVID_Na_df[, -c(1, 2, 3, 4)], as.numeric)
```

```r
COVID_Na_df2 <-
    data.frame("country_name" = COVID_Na_df$country_name,
               "dpm_100d" = COVID_Na_df$deaths_per_million_100_days_after_fifth_death,
               "si_100d" = COVID_Na_df$stringency_index_100_days_after_fifth_death)
##############################################################################################

# Juntamos ambos dataframes en uno sólo.
COVID_BGC <-
    inner_join(BCG_no_strain_simple, COVID_Na_df2, by = "country_name")
```

Nuestra tabla resultante es la siguiente:

Table 2: Tabla 1. Vacunación de BCG por países y muertes por COVID-19

| country_name | periods_with_vaccine | vaccination_2020_2015 | last_vaccine_year | first_vaccine_year | dpm_100d |
|---|---|---|---|---|---|
| Afghanistan | 6 | 1 | 2020 | 1980 | 54 |
| Angola | 6 | 1 | 2020 | 1980 | 99 |
| Argentina | 3 | 0 | 2010 | 1990 | 79 |
| Armenia | 4 | 1 | 2020 | 2000 | 37 |
| Australia | 9 | 1 | 2020 | 1950 | 94 |
| Bangladesh | 6 | 1 | 2020 | 1980 | 12 |

Mediante esta tabla llevaremos a cabo nuestros análisis. A continuación mostramos la estructura de la misma:

```r
str(COVID_BGC)
```

```
## 'data.frame':    65 obs. of  7 variables:
##  $ country_name         : Factor w/ 222 levels "Afghanistan",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ periods_with_vaccine : num  6 6 3 4 9 6 6 9 8 9 ...
##  $ vaccination_2020_2015: int  1 1 0 1 1 1 1 1 1 1 ...
##  $ last_vaccine_year    : num  2020 2020 2010 2020 2020 2020 2020 2020 2020 2020 ...
##  $ first_vaccine_year   : num  1980 1980 1990 2000 1950 1980 1980 1950 1960 1950 ...
##  $ dpm_100d             : num  54 99 79 37 94 12 NA 128 57 88 ...
##  $ si_100d              : num  68 NA 76 NA 30 61 NA 34 66 16 ...
```

Podemos ver que nuestro data frame tiene 65 observacones y 7 columnas. Las columnas son:

- **country_name**: El nombre del país en cuestión.
- **periods_with_vaccine**: el número de periodos estudiados con vacunación activa.
- **vaccination_2020_2015**: el único periodo considerado en este estudio (el último).
- **last_vaccine_year**: de los años estudiados, el último con campaña de vacunación. En el caso de no tener vacunación, este será el primer año estudiado (1950).
- **first_vaccine_year**: de los años estudiados, el primero con campaña de vacunación. En el caso de no tener vacunación, este será el último año estudiado (2020).
- **dpm_100d**: muertes por millón tras haber pasado 100 días desde la quinta muerte registrada.
- **si_100d**: "stringency index" (indicador que va de 0 a 100 que ide la severidad de las medidas tomadas por el país para aplacar la pandemia) tras haber pasado 100 días desde la quinta muerte registrada.

**Hagamos unos análisis descriptivos:** **a)** ¿Existen valores nulos en el conjunto de datos?

```r
table(is.null(COVID_BGC))
```

```
##
## FALSE
```

```
##     1
```

No existen valores nulos.

**b)** ¿Existen *"missing values"* en el conjunto de datos?

```
table(is.na(COVID_BGC))
```

```
##
## FALSE  TRUE
##   434    21
```

Tenemos 21 países con algún valor perdido. Hemos de tener esto en cuenta para futuros análisis.

**Hagamos un poco de estadística descriptiva:** **a)** Resumen estadístico de las variables

```
# Obvimente hay variables en las que no tiene sentido hacer resumen estadístico,
# como el alpha_3_code, las strains... Pero por ahora lo voy a dejar
summary(COVID_BGC)
```

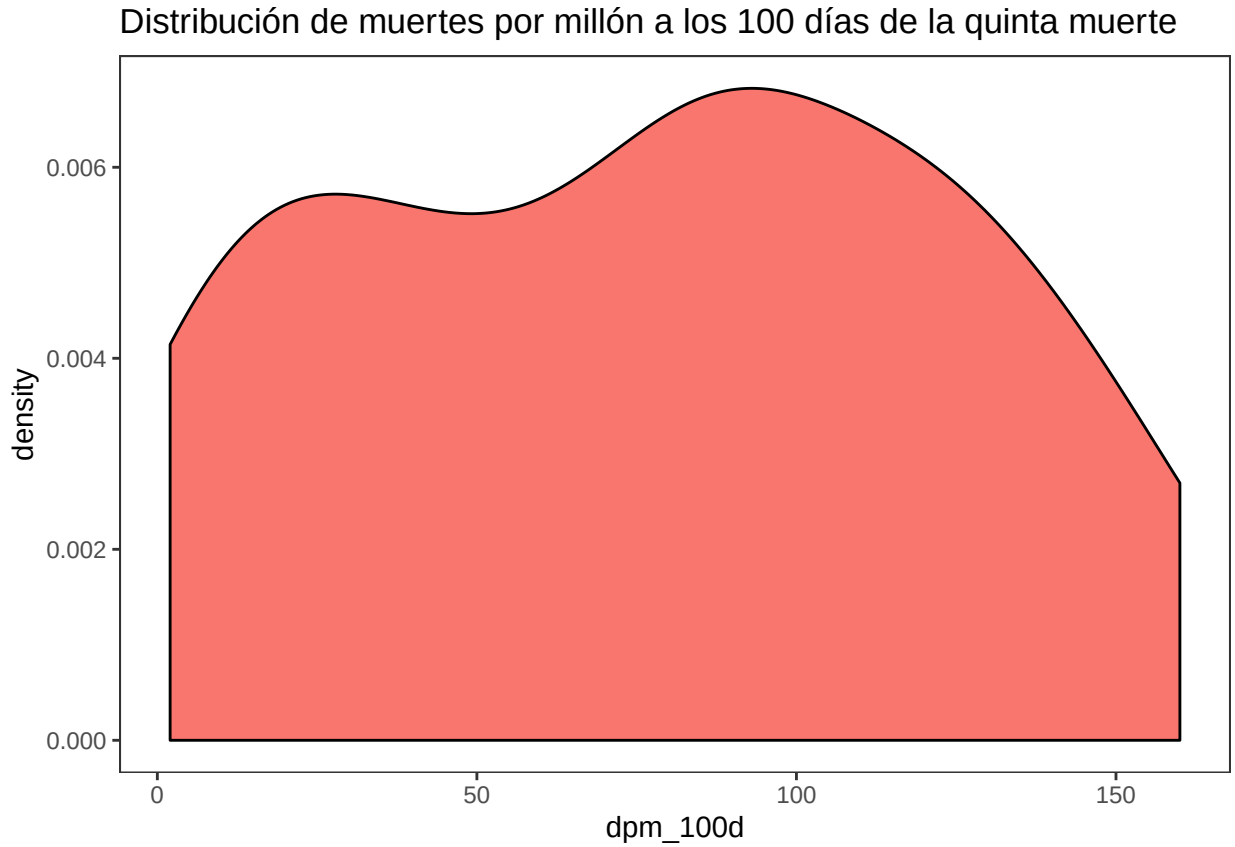
```
##       country_name periods_with_vaccine vaccination_2020_2015 last_vaccine_year
##  Afghanistan: 1    Min.   :0.000        Min.   :0.0000        Min.   :1950
##  Angola     : 1    1st Qu.:5.000        1st Qu.:1.0000        1st Qu.:2020
##  Argentina  : 1    Median :6.000        Median :1.0000        Median :2020
##  Armenia    : 1    Mean   :6.277        Mean   :0.7538        Mean   :2012
##  Australia  : 1    3rd Qu.:8.000        3rd Qu.:1.0000        3rd Qu.:2020
##  Bangladesh : 1    Max.   :9.000        Max.   :1.0000        Max.   :2020
##  (Other)    :59
##  first_vaccine_year    dpm_100d         si_100d
##  Min.   :1950       Min.   :  2.00   Min.   : 3.00
##  1st Qu.:1950       1st Qu.: 35.75   1st Qu.:19.00
##  Median :1960       Median : 80.00   Median :39.00
##  Mean   :1969       Mean   : 76.95   Mean   :40.09
##  3rd Qu.:1980       3rd Qu.:118.00   3rd Qu.:60.00
##  Max.   :2020       Max.   :160.00   Max.   :76.00
##                     NA's   :9        NA's   :12
```

**b)** Distribución de las variables

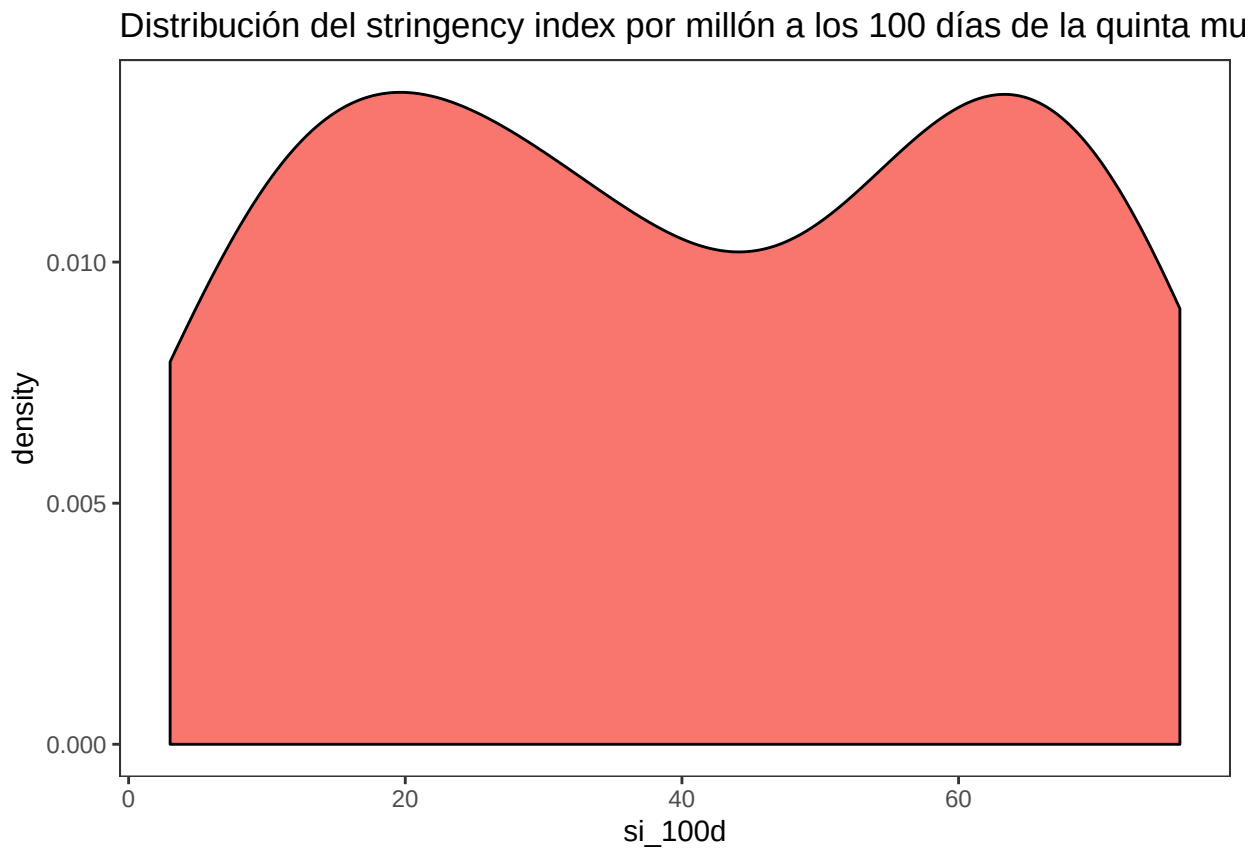
Observemos cómo se distribuyen los datos de *"deaths per million 100 days after fifth death"* y de *"stringency index 100 days after fifth death"*

Mediante el gráfico de densidad podemos ver la distribución de dos variables continuas como son el número de muertes por millón y el stringency index.

```
ggplot(COVID_BGC, aes(x= dpm_100d, fill = "b"))+geom_density()+
  theme_bw()+theme(legend.position = 0, panel.grid = element_blank())+
  ggtitle("Distribución de muertes por millón a los 100 días de la quinta muerte")
```

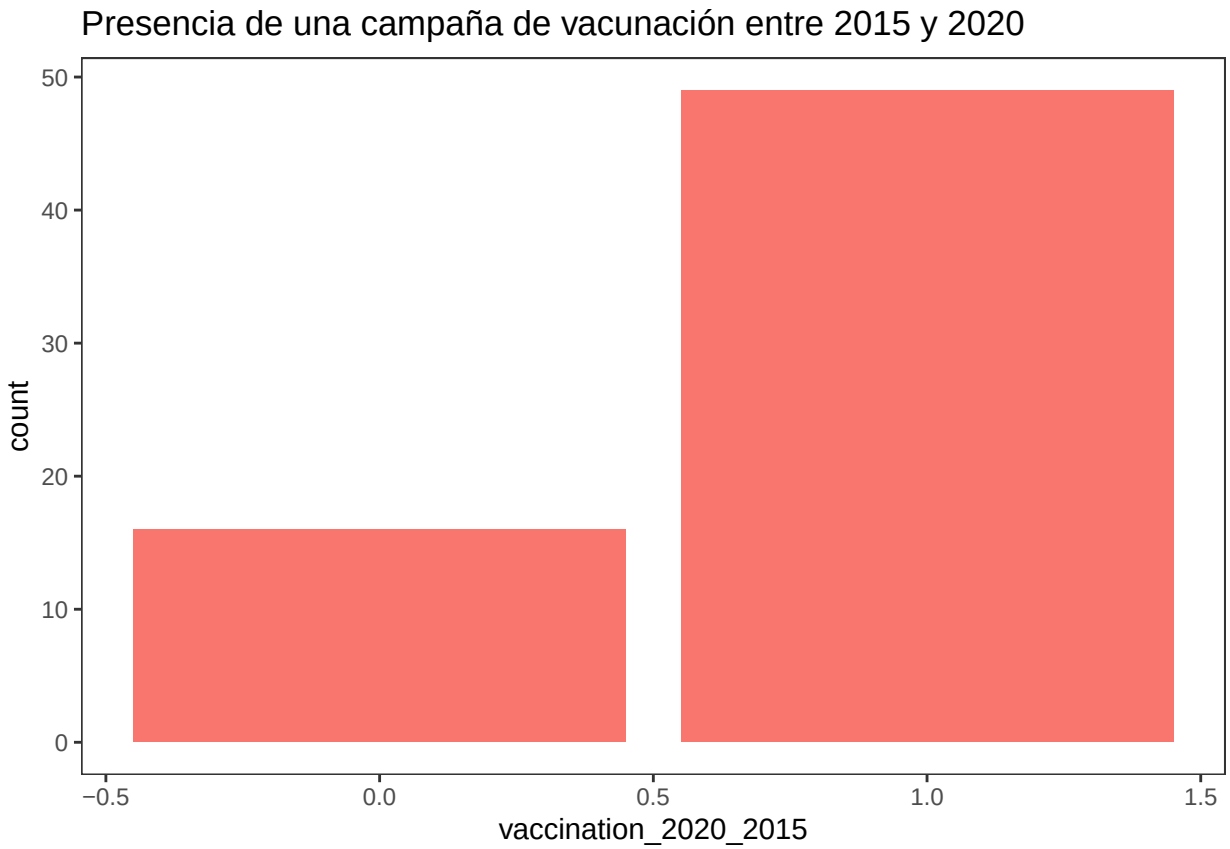



```
ggplot(COVID_BGC, aes(x= si_100d, fill = "b"))+geom_density()+
  theme_bw()+theme(legend.position = 0, panel.grid = element_blank())+
  ggtitle("Distribución del stringency index por millón a los 100 días de la quinta muerte")
```



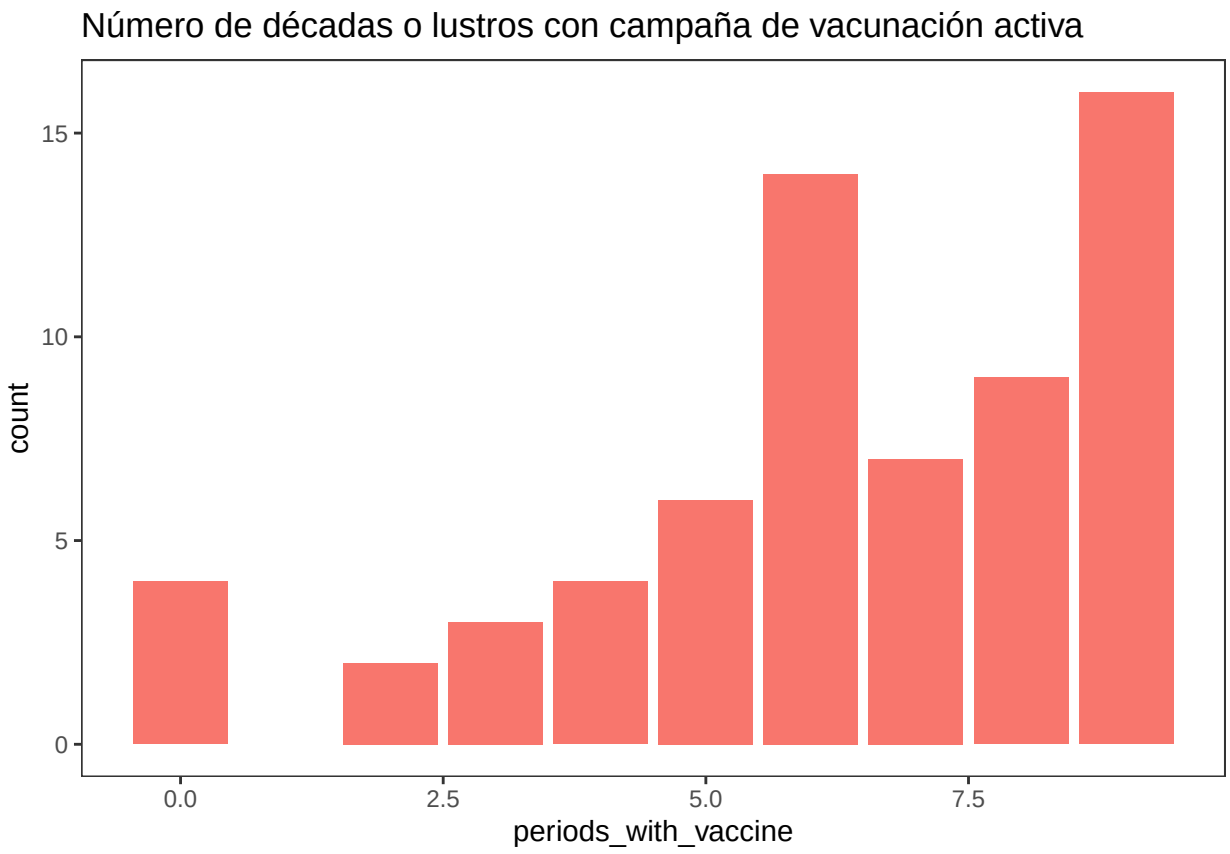

Podemos comprobar la distribución de una variable categórica como es la presencia o no de campaña de vacunación entre 2015 y 2020 con un gráfico de barras.

```
ggplot(COVID_BGC, aes(x= vaccination_2020_2015, fill = "b"))+geom_bar()+
  theme_bw()+theme(legend.position = 0, panel.grid = element_blank())+
  ggtitle("Presencia de una campaña de vacunación entre 2015 y 2020")
```



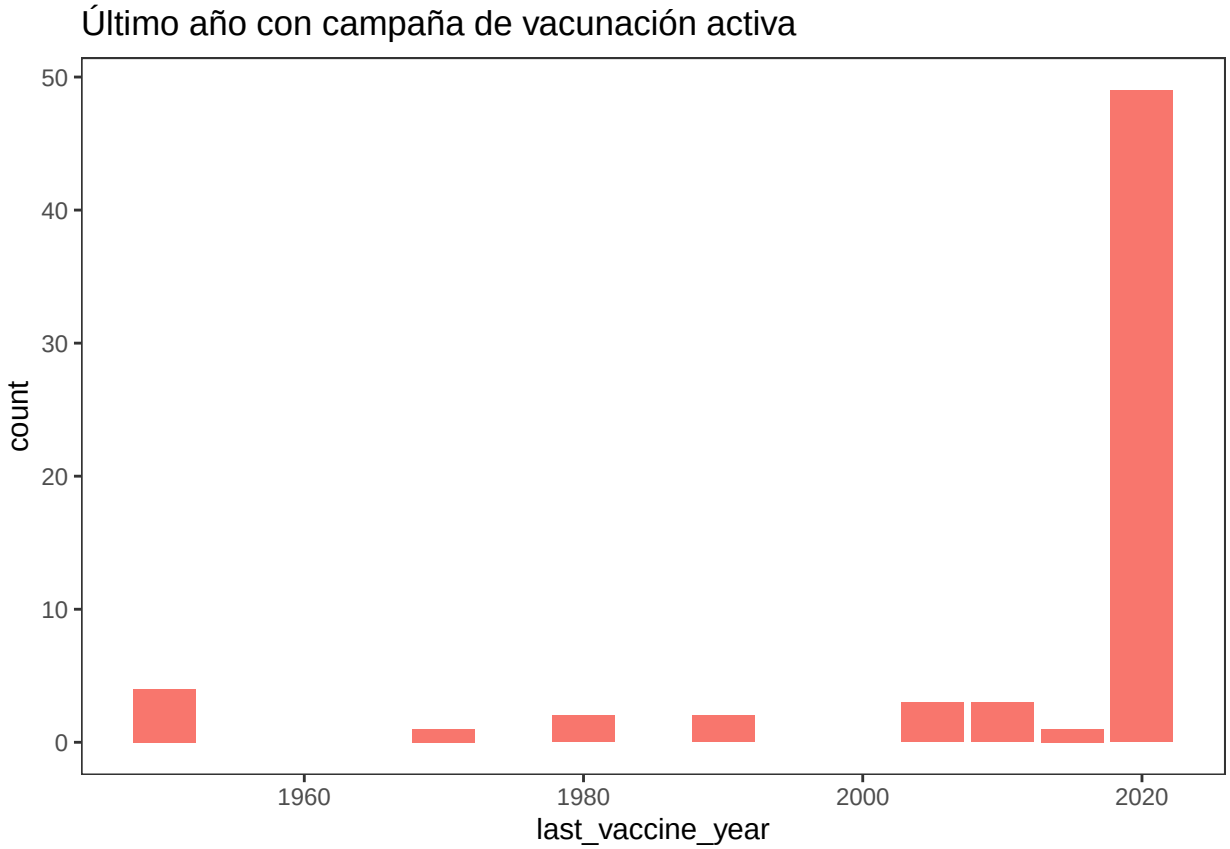

También podemos comprobar qué distribución sigue una variable binomial, es decir, el número de periodos con vacuna de 9 periodos posibles. Esta es una variable binomial ya que es resultado de n=9 observaciones con resultado de éxito (hay vacunación) o fracaso (no lo hay), Para observar su distribución podemos utilizar un histograma o un gráfico de barras.

```
ggplot(COVID_BGC, aes(x= periods_with_vaccine, fill = "b"))+geom_bar()+
  theme_bw()+theme(legend.position = 0, panel.grid = element_blank())+
  ggtitle("Número de décadas o lustros con campaña de vacunación activa")
```

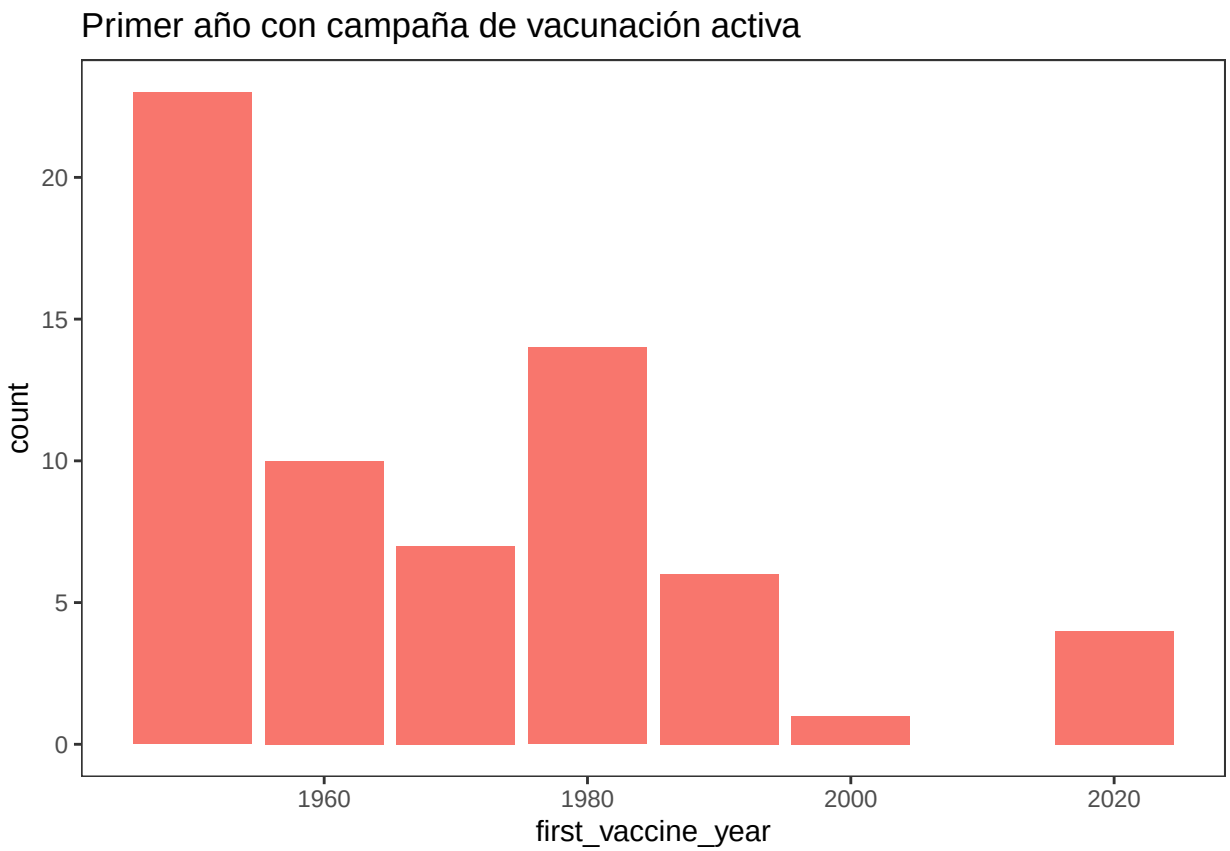



Dos variables muy diferentes son las que se refieren al primer y último año de vacunación. Al tratarse de años, podrían seguir una distribución uniforme. Pero es lógico pensar que estos puedan tener una distribución simétrica semejante a la normal, ya que los países tienden a comenzar campañas de vacunación en momentos parecidos. Veámos qué distribución tienen los datos con un gráfico de barras.

```
ggplot(COVID_BGC, aes(x=last_vaccine_year, fill = "b"))+geom_bar()+
  theme_bw()+theme(legend.position = 0, panel.grid = element_blank())+
  ggtitle("Último año con campaña de vacunación activa")
```




```
ggplot(COVID_BGC, aes(x=first_vaccine_year, fill = "b"))+geom_bar()+
  theme_bw()+theme(legend.position = 0, panel.grid = element_blank())+
  ggtitle("Primer año con campaña de vacunación activa")
```




Para trabajar los datos de manera efectiva y siguiendo la metodología de la *PEC 2*, escribo funciones específicas para obtener información de nuestro conjunto de datos.

Por ejemplo, desarrollo una función para obtener el último año en el que se administró la vacuna BCG en un determinado país.

```
get_last_vaccine_year <- function(country){
  # Devuelve el último año con vacuna de un país country. Argumentos:
  # Country = string indicando el nombre del país

  # Comprobamos si la string pertenece a los paises estudiados.
  # Si no, imprimimos un mensaje y detenemos la función.
  if (sapply(COVID_BGC$country_name, function(x){x==country})%>% sum()==0){
    print("El país especificado no está en la lista")
  }else{
    # Si está incluida, obtengo su last vaccine year.
    COVID_BGC[COVID_BGC$country_name == country,]$last_vaccine_year
  }
}
get_last_vaccine_year("Spain")
```

```
## [1] 1980
```

```
get_last_vaccine_year("URSS") # Comprobamos que da error, al no estar incluído
```

```
## [1] "El país especificado no está en la lista"
```

De manera parecida, podemos tratar de comprobar si, en un año determinado, hubo campaña de vacunación en un determinado país.

```r
was_there_a_vaccine <- function(country, year){
  # Devuelve un valor lógico dependiendo de si el año especificado hubo o no vacuna en un país.
  # Argumentos:
  # Country = string indicando el nombre del país
  # year = valor numérico indicando el año en duda


  #En este caso, comrpobamos si el país está en la lista de paises y si el año está
  # incluido en los años estudiados.
  if (sapply(COVID_BGC$country_name, function(x){x==country})%>% sum()==0){
    print("El país especificado no está en la lista")
  }else if(year > max(COVID_BGC$last_vaccine_year) | year < min(COVID_BGC$first_vaccine_year) ){
    print("El año especificado está fuera del rango")
  }else{
    # Si todo se cumple, estudiaremos si el año se encuentra en el periodo en el
    # que hubo vacunación.
    last <- COVID_BGC[COVID_BGC$country_name == country,]$last_vaccine_year
    first <- COVID_BGC[COVID_BGC$country_name == country,]$first_vaccine_year
    if (last > year && year > first){
      return(TRUE)
    }else{
      return(FALSE)
    }
  }
}

was_there_a_vaccine(country = "Spain", year = 1975)
```

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

```r
was_there_a_vaccine("URSS", 1960)
```

```
## [1] "El país especificado no está en la lista"
```

```r
was_there_a_vaccine("Spain", 2022)
```

```
## [1] "El año especificado está fuera del rango"
```

**c)** ¿Cuál es la media para la variable `dpm_100d` para el conjunto de países que la han medido? ¿Y para la variable `si_100d`?

```r
m1 <- mean(na.omit(COVID_BGC$dpm_100d))
m2 <- mean(na.omit(COVID_BGC$si_100d))
cat(sprintf("La media para dpm_100d es %.2f\n", m1))
```

```
## La media para dpm_100d es 76.95
```

```r
cat(sprintf("La media para si_100d es %.2f\n", m2))
```

```
## La media para si_100d es 40.09
```

Un detalle importante que se puede apreciar es que la media para el **"stringency index"** de los países considerados en este conjunto de datos se encuentra por debajo de 50, lo que significaría que, de media, las medidas tomadas por los países para aplacar la pandemia 100 días tras la quinta muerte registrada no fueron muy severas.

**d)** ¿Cuáles son la varianza y la desviazon estandar para la variable `dpm_100d` para el conjunto de países que las han medido? ¿Y para la variable `si_100d`?

```r
v1 <- var(na.omit(COVID_BGC$dpm_100d))
v2 <- var(na.omit(COVID_BGC$si_100d))
sd1 <- sd(na.omit(COVID_BGC$dpm_100d))
sd2 <- sd(na.omit(COVID_BGC$si_100d))

cat(sprintf("La varianza para dpm_100d es %.2f\n", v1))
```

```
## La varianza para dpm_100d es 2243.62
```

```r
cat(sprintf("La desviación estandar para dpm_100d es %.2f\n", sd1))
```

```
## La desviación estandar para dpm_100d es 47.37
```

```r
cat(sprintf("La varianza para si_100d es %.2f\n", v2))
```

```
## La varianza para si_100d es 552.05
```

```r
cat(sprintf("La desviación estandar para si_100d es %.2f\n", sd2))
```

```
## La desviación estandar para si_100d es 23.50
```

**e)** ¿Cuáles son los países cuyo valor para `dpm_100d` (en caso de estar presente) se encuentra por debajo de la media? ¿Y para la variable `si_100d`?

```r
cat("Los países cuyo valor de dpm_100d es menor que la media del dataset son:\n\n")
```

```
## Los países cuyo valor de dpm_100d es menor que la media del dataset son:
```

```r
targets <- as.character(subset(COVID_BGC, si_100d < m2)$country_name)
if (length(targets) %% 8 != 0){
  t = (length(targets) %/% 8) + 1
} else {
  t = (length(targets) %/% 8)
}

for (i in 1:t) {
  cat("\n")
  out <- targets[(8*i-7):(8*i)]
  cat(paste(out[!is.na(out)], collapse = ', '), "\n")
}
```

```
##
## Australia, Bosnia and Herzegovina, Bulgaria, Czech Republic, Estonia, Finland, Greece, Guam
##
## Hungary, Indonesia, Ireland, Italy, Japan, Latvia, Malaysia, Netherlands
##
## Pakistan, Poland, Senegal, Sierra Leone, Spain, Sweden, Switzerland, Taiwan
##
## Tanzania, Thailand, Tunisia, Ukraine, Uruguay
```

```r
cat("\n\n\nLos países cuyo valor de si_100d es menor que la media del dataset son:\n\n")
```

```
##
##
##
## Los países cuyo valor de si_100d es menor que la media del dataset son:
```

```r
targets <- as.character(subset(COVID_BGC, si_100d < m2)$country_name)
if (length(targets) %% 8 != 0){
```

```
  t = (length(targets) %/% 8) + 1
} else {
  t = (length(targets) %/% 8)
}

for (i in 1:t) {
  cat("\n")
  out <- targets[(8*i-7):(8*i)]
  cat(paste(out[!is.na(out)], collapse = ', '), "\n")
}
```

```
##
## Australia, Bosnia and Herzegovina, Bulgaria, Czech Republic, Estonia, Finland, Greece, Guam
##
## Hungary, Indonesia, Ireland, Italy, Japan, Latvia, Malaysia, Netherlands
##
## Pakistan, Poland, Senegal, Sierra Leone, Spain, Sweden, Switzerland, Taiwan
##
## Tanzania, Thailand, Tunisia, Ukraine, Uruguay
```

**f)** ¿Cuáles son los países que cumplen ambas condiciones del apartado anterior?

```
cat(
  "Los países cuyos valores de dpm_100d y de si_100d son menores que la media del dataset son:\n\n",
  paste(subset(COVID_BGC, (dpm_100d < m1) &
                 (si_100d < m2))$country_name, collapse = ', ')
)
```

```
## Los países cuyos valores de dpm_100d y de si_100d son menores que la media del dataset son:
##
##  Greece, Guam, Latvia, Pakistan, Senegal, Switzerland, Taiwan, Tanzania, Thailand, Ukraine
```

**g)** ¿Cuáles son los países que han tenido campaña de vacunación de la vacuna *BCG* más reciente y que su la media de mortalidad a los 100 días es menor que la media?

```
cat(
  paste(
    "Los países cuyos valores de dpm_100d son menores que la media del\n",
    "dataset y que además han tenido una reciente campaña de vacunación\n",
    "de la vacuna BCG son:\n\n"
  )
)
```

```
## Los países cuyos valores de dpm_100d son menores que la media del
##  dataset y que además han tenido una reciente campaña de vacunación
##  de la vacuna BCG son:
```

```
targets <- as.character(subset(COVID_BGC, (dpm_100d < m1) &
                                 (vaccination_2020_2015 == 1))$country_name)

if (length(targets) %% 8 != 0){
  t = (length(targets) %/% 8) + 1
} else {
  t = (length(targets) %/% 8)
}

for (i in 1:t) {
```

```
  cat("\n")
  out <- targets[(8*i-7):(8*i)]
  cat(paste(out[!is.na(out)], collapse = ', '), "\n")
}
```

```
##
## Afghanistan, Armenia, Bangladesh, Brazil, India, Kazakhstan, Kuwait, Latvia
##
## Malta, Mexico, Nigeria, Pakistan, Peru, Philippines, Portugal, Senegal
##
## Sudan, Taiwan, Tanzania, Thailand, Ukraine
```

**h)** ¿Cuáles son las frecuencias relativas y absolutas de la variable `si_100d`?

```
prop.table(table(COVID_BGC$si_100d))
```

```
##
##          3          4          5          7          8         12         15
## 0.01886792 0.01886792 0.01886792 0.03773585 0.01886792 0.05660377 0.01886792
##         16         17         18         19         22         24         26
## 0.01886792 0.01886792 0.01886792 0.03773585 0.01886792 0.03773585 0.01886792
##         28         30         31         34         36         38         39
## 0.03773585 0.01886792 0.01886792 0.01886792 0.01886792 0.01886792 0.01886792
##         40         43         47         51         55         56         57
## 0.03773585 0.01886792 0.01886792 0.01886792 0.01886792 0.05660377 0.01886792
##         60         61         65         66         67         68         69
## 0.05660377 0.01886792 0.01886792 0.01886792 0.01886792 0.01886792 0.01886792
##         70         71         73         74         75         76
## 0.01886792 0.01886792 0.01886792 0.01886792 0.03773585 0.01886792
```

**i)** ¿Cuáles son las frecuencias relativas y absolutas de la variable `dpm_100d`?

```
prop.table(table(COVID_BGC$dpm_100d))
```

```
##
##          2          3          5          7          8          9         11
## 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714
##         12         13         23         26         27         33         35
## 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714
##         36         37         39         45         50         54         57
## 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714
##         61         63         67         76         77         78         79
## 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714
##         81         82         88         89         90         91         94
## 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714
##         97         99        103        106        107        109        117
## 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714
##        121        122        124        125        127        128        132
## 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714
##        134        139        148        152        153        158        160
## 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714 0.01785714
```

**j)** ¿Cuáles son las frecuencias relativas y absolutas de la variable `vaccination_2020_2015`?

```
prop.table(table(COVID_BGC$vaccination_2020_2015))
```

```
##
```

```
##         0         1
## 0.2461538 0.7538462
```

**k)** ¿Cuáles son las frecuencias relativas cruzadas de las variables `dpm_100d` y `vaccination_2020_2015`?

```r
prop.table(table(COVID_BGC$dpm_100d, COVID_BGC$vaccination_2020_2015))
```

```
##
##                0          1
##   2   0.00000000 0.01785714
##   3   0.00000000 0.01785714
##   5   0.00000000 0.01785714
##   7   0.00000000 0.01785714
##   8   0.00000000 0.01785714
##   9   0.01785714 0.00000000
##   11  0.00000000 0.01785714
##   12  0.00000000 0.01785714
##   13  0.00000000 0.01785714
##   23  0.00000000 0.01785714
##   26  0.00000000 0.01785714
##   27  0.00000000 0.01785714
##   33  0.01785714 0.00000000
##   35  0.00000000 0.01785714
##   36  0.00000000 0.01785714
##   37  0.00000000 0.01785714
##   39  0.01785714 0.00000000
##   45  0.00000000 0.01785714
##   50  0.00000000 0.01785714
##   54  0.00000000 0.01785714
##   57  0.00000000 0.01785714
##   61  0.00000000 0.01785714
##   63  0.00000000 0.01785714
##   67  0.01785714 0.00000000
##   76  0.00000000 0.01785714
##   77  0.00000000 0.01785714
##   78  0.00000000 0.01785714
##   79  0.01785714 0.00000000
##   81  0.01785714 0.00000000
##   82  0.00000000 0.01785714
##   88  0.00000000 0.01785714
##   89  0.01785714 0.00000000
##   90  0.01785714 0.00000000
##   91  0.00000000 0.01785714
##   94  0.00000000 0.01785714
##   97  0.00000000 0.01785714
##   99  0.00000000 0.01785714
##   103 0.00000000 0.01785714
##   106 0.01785714 0.00000000
##   107 0.00000000 0.01785714
##   109 0.00000000 0.01785714
##   117 0.01785714 0.00000000
##   121 0.01785714 0.00000000
##   122 0.00000000 0.01785714
##   124 0.01785714 0.00000000
##   125 0.01785714 0.00000000
```

```
##   127 0.01785714 0.00000000
##   128 0.00000000 0.01785714
##   132 0.00000000 0.01785714
##   134 0.00000000 0.01785714
##   139 0.00000000 0.01785714
##   148 0.00000000 0.01785714
##   152 0.00000000 0.01785714
##   153 0.01785714 0.00000000
##   158 0.00000000 0.01785714
##   160 0.00000000 0.01785714
```

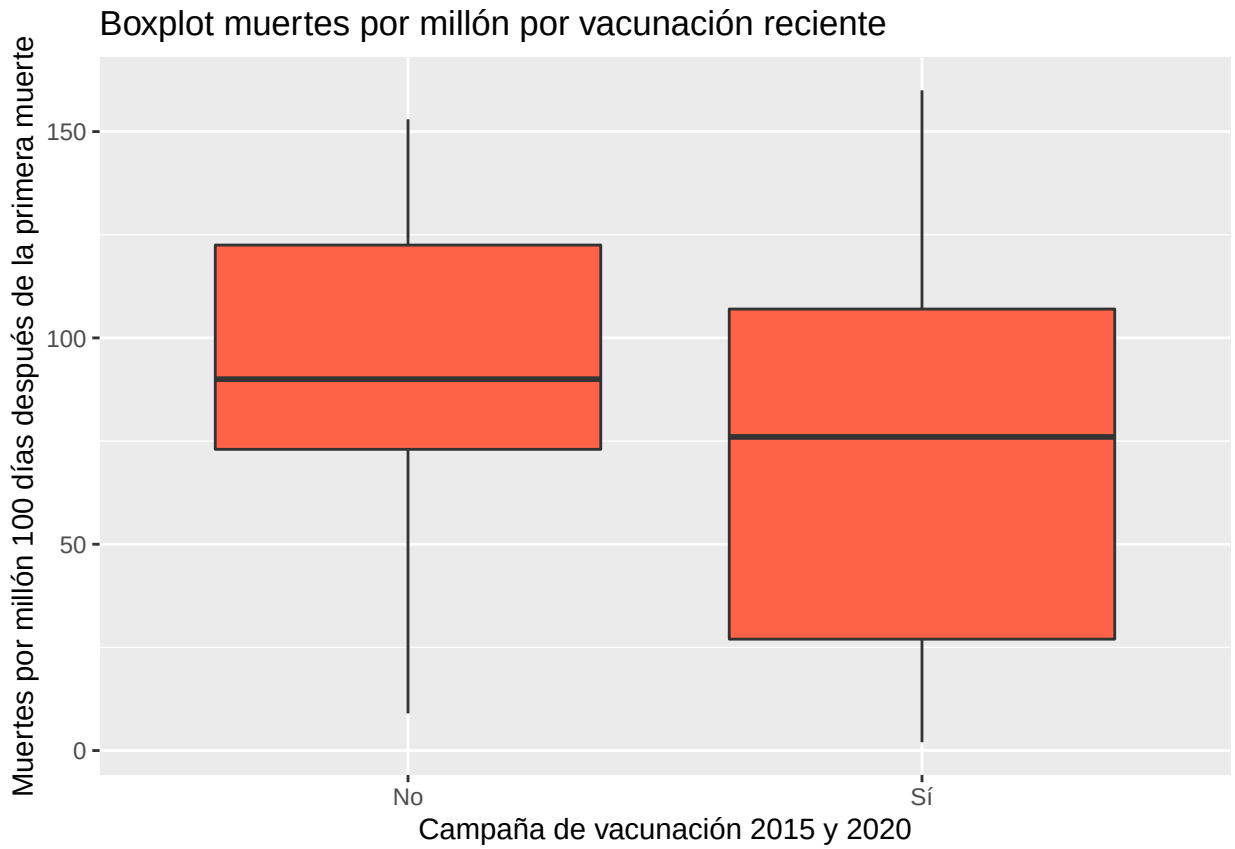
**l)** Hagamos un diagrama de tallo y hojas de la variable `dpm_100d`.

```
stem(COVID_BGC$dpm_100d)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##    0 | 235789123
##    2 | 36735679
##    4 | 5047
##    6 | 1376789
##    8 | 128901479
##   10 | 36797
##   12 | 124578249
##   14 | 8238
##   16 | 0
```

**m)** Hagamos unos diagramas de cajas y bigotes para la variable `dpm_100d` argupada según los valores de `vaccination_2020_2015`.

```
bxp_COVID_BGC <- COVID_BGC
bxp_COVID_BGC$vaccination_2020_2015 <- factor(
                                              bxp_COVID_BGC$vaccination_2020_2015,
                                              labels = c("No", "Sí")
                                              )
(
 ggplot(
        bxp_COVID_BGC,
        aes(x = vaccination_2020_2015, y = dpm_100d)
        )
        + geom_boxplot(fill = "tomato1")
        + labs(
               title = "Boxplot muertes por millón por vacunación reciente",
               x = "Campaña de vacunación 2015 y 2020",
               y = "Muertes por millón 100 días después de la primera muerte"
               )
 )
```




## Estudio de los datos: Machine learning

En el siguiente paso, estudiaremos los datos presentados, intentando demostrar alguna relación entre las variables estudiadas.

### Correlación

Como primer paso, buscaremos relaciones lineales entre las variables. Para ello, podemos computar una matriz de correlaciones entre las variables. Si encontramos alguna correlación podemos utilizarla para realizar un modelo de regresión lineal.

Para comenzar, computamos la matriz de correlaciones, omitiendo los valores NA y seleccionando todas las variables menos el nombre del país.

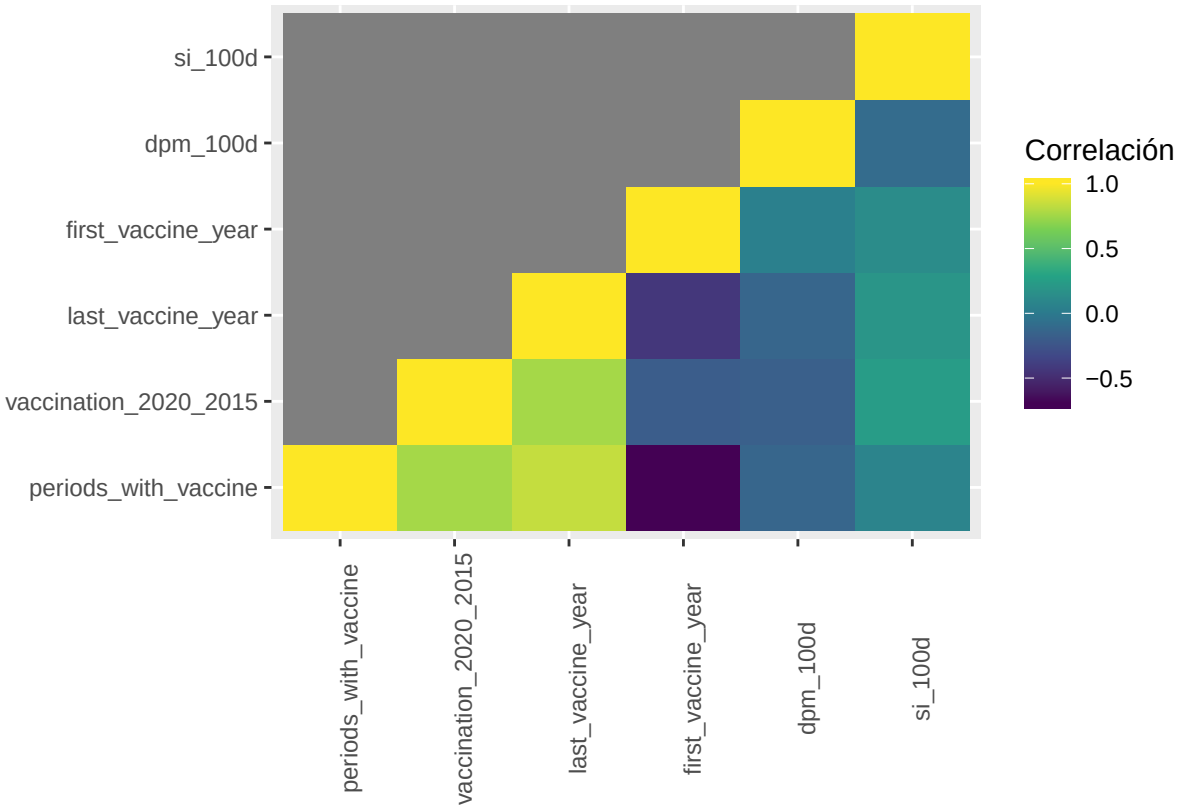
```
df <- COVID_BGC[2:ncol(COVID_BGC)] %>% na.omit()
cormat <-
    cor(df)
cormat
```

```
##                       periods_with_vaccine vaccination_2020_2015
## periods_with_vaccine            1.00000000             0.7589950
## vaccination_2020_2015           0.75899495             1.0000000
## last_vaccine_year               0.84135869             0.7642953
## first_vaccine_year             -0.68594025            -0.1901543
## dpm_100d                       -0.13017502            -0.1685687
## si_100d                         0.07406476             0.2401127
##                       last_vaccine_year first_vaccine_year    dpm_100d
## periods_with_vaccine          0.8413587        -0.68594025 -0.13017502
```

```
## vaccination_2020_2015         0.7642953        -0.19015427 -0.16856874
## last_vaccine_year             1.0000000        -0.43202170 -0.13167163
## first_vaccine_year           -0.4320217         1.00000000  0.04061753
## dpm_100d                     -0.1316716         0.04061753  1.00000000
## si_100d                       0.1934466         0.12981251 -0.08834297
##                            si_100d
## periods_with_vaccine    0.07406476
## vaccination_2020_2015   0.24011270
## last_vaccine_year       0.19344665
## first_vaccine_year      0.12981251
## dpm_100d               -0.08834297
## si_100d                 1.00000000
```

Y visualizamos la matriz usando ggplot geom_tile.

```
cormat2 <- cormat
cormat2[upper.tri(cormat2)] <-
    NA #Para visualizar solamente una vez las correlaciones
cormat2 <- melt(round(cormat2, 2)) #Formato para poder usar ggplot
(
ggplot(cormat2, aes(x = Var1, y = Var2, fill = value))
  + geom_tile() + scale_fill_continuous(type = "viridis")
  + theme(axis.text.x = element_text(angle = 90))
  + xlab("") + ylab("") + labs(fill = "Correlación")
)
```




En el anterior gráfico vemos diferentes variables relacionadas de manera lineal. Algunas de las relaciones se deben a que son variables relacionadas y, por lo tanto, no nos aportan nueva información ya que son

relaciones ya conocidas. Como las relaciones entre la vacunación en 2020-2015 y el número de periodos con vacuna o el último año de vacunación; o, en definitiva, cualquier relación entre las variables que tratan de los años de vacunación.

Las relaciones que buscamos son aquellas que relacionen algún dato sobre la campaña de vacunación y el número de muertes por millón de habitantes. Para este caso, vemos que existen correlaciones negativas entre el número de muertes y el último año de vacunación, además de con la vacunación entre 2020 y 2015.

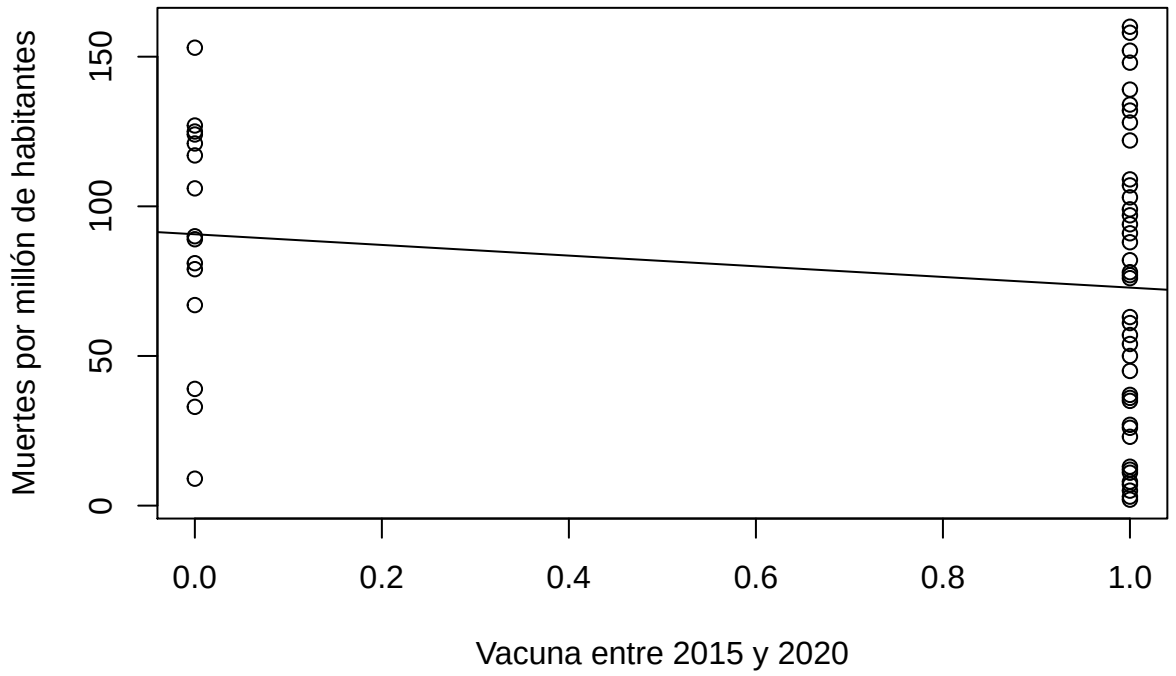
Revisamos los valores anteriores podemos ver que estas correlaciones son de -0.55 (dpm_100d, vaccination_2020_2015) y -0.6 (dpm_100d, last_vaccine_year).

### Regresión lineal

Con los datos anteriores podemos pasar a construir un modelo de regresión lineal en el que la variable dependiente sea el número de muertes y las variables dependientes sean las dos variables mencionadas antes.

En primer lugar, utilizando la vacunación entre 2015 y 2020.

```
modelo <- lm(dpm_100d ~ vaccination_2020_2015, data = df)
plot(
     COVID_BGC$vaccination_2020_2015,
     COVID_BGC$dpm_100d,
     xlab="Vacuna entre 2015 y 2020",
     ylab="Muertes por millón de habitantes"
     )
abline(lm(dpm_100d ~ vaccination_2020_2015, df))
```

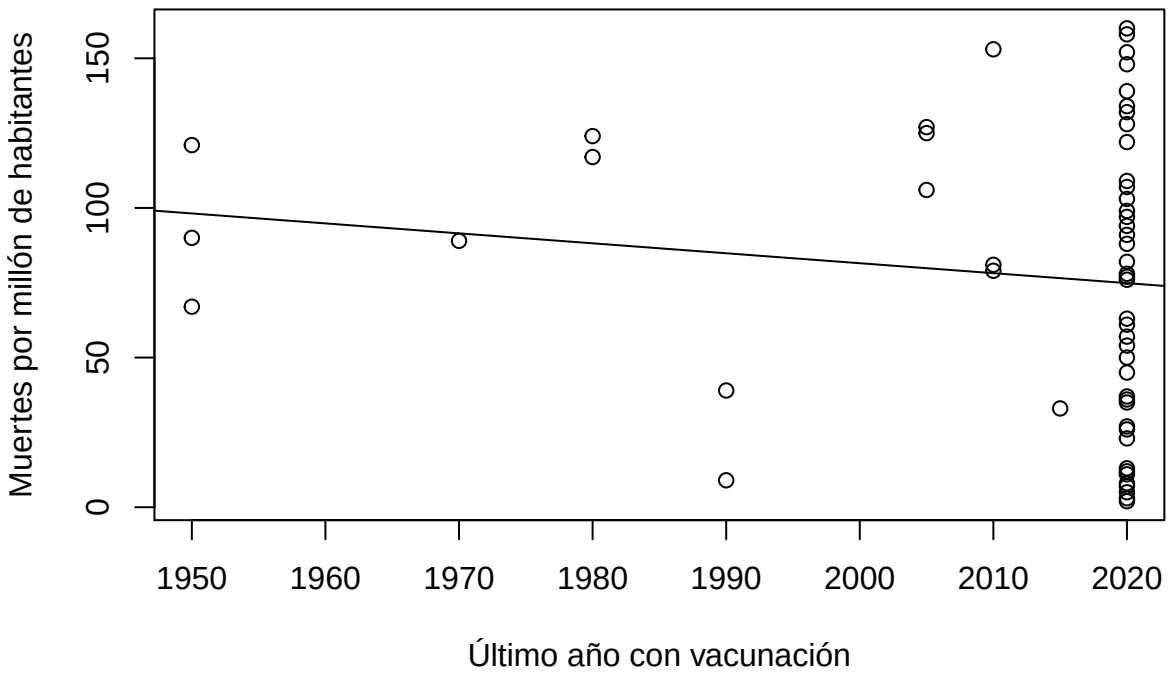



```
summary(modelo)
```

```
##
## Call:
## lm(formula = dpm_100d ~ vaccination_2020_2015, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -81.667 -45.842   3.158  34.158  87.158
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)              90.67      12.36   7.337  1.6e-09 ***
## vaccination_2020_2015   -17.82      14.59  -1.221    0.228
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.86 on 51 degrees of freedom
## Multiple R-squared:  0.02842,    Adjusted R-squared:  0.009365
## F-statistic: 1.492 on 1 and 51 DF,  p-value: 0.2276
```

En segundo lugar, el último año de vacunación.

```
modelo <- lm(dpm_100d ~ last_vaccine_year,data = df)
plot(
     COVID_BGC$last_vaccine_year,
     COVID_BGC$dpm_100d,
     xlab="Último año con vacunación",
     ylab="Muertes por millón de habitantes"
     )
abline(lm(dpm_100d ~ last_vaccine_year, df))
```




```
summary(modelo)
```

```
##
## Call:
## lm(formula = dpm_100d ~ last_vaccine_year, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.851 -43.538   2.125  34.125  85.125
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       746.6080   705.0000   1.059    0.295
## last_vaccine_year  -0.3325     0.3506  -0.949    0.347
##
## Residual standard error: 48.13 on 51 degrees of freedom
## Multiple R-squared:  0.01734,    Adjusted R-squared:  -0.00193
## F-statistic: 0.8998 on 1 and 51 DF,  p-value: 0.3473
```

Vemos que el último modelo es mejor, ya que tiene un menor error y un mayor R cuadrado.

Podemos construir un modelo múltiple también, utilizand ambos predictores.

```
modelo <- lm(dpm_100d ~ last_vaccine_year+vaccination_2020_2015 ,data = df)
summary(modelo)
```

```
##
## Call:
```

```
## lm(formula = dpm_100d ~ last_vaccine_year + vaccination_2020_2015,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -81.632 -45.842   3.158  34.158  87.158
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            124.89881 1085.37388   0.115    0.909
## last_vaccine_year       -0.01722    0.54593  -0.032    0.975
## vaccination_2020_2015  -17.27354   22.85720  -0.756    0.453
##
## Residual standard error: 48.34 on 50 degrees of freedom
## Multiple R-squared:  0.02843,    Adjusted R-squared:  -0.01043
## F-statistic: 0.7317 on 2 and 50 DF,  p-value: 0.4862
```

En este modelo observamos un ligero incremento en R cuadrado y que el efecto del predictor vaccination_2020_2015 no es significativo. Podemos considerar, por lo tanto, que el modelo que incluye sólo la variable last_vaccine_year es mejor.

Como última comporbación, podemos observar un modelo que incluya todas las variables del conjunto de datos.

```
modelo <- lm(dpm_100d ~ ., data = df)
summary(modelo)
```

```
##
## Call:
## lm(formula = dpm_100d ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -78.900 -46.136   2.675  33.542  88.310
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -74.49935 1868.52742  -0.040    0.968
## periods_with_vaccine     0.24060    9.27347   0.026    0.979
## vaccination_2020_2015  -17.34337   33.22089  -0.522    0.604
## last_vaccine_year        0.01409    0.73819   0.019    0.985
## first_vaccine_year       0.07087    0.72162   0.098    0.922
## si_100d                 -0.11163    0.31153  -0.358    0.722
##
## Residual standard error: 49.79 on 47 degrees of freedom
## Multiple R-squared:  0.03122,    Adjusted R-squared:  -0.07184
## F-statistic: 0.3029 on 5 and 47 DF,  p-value: 0.9087
```

Vemos, por lo tanto, que este es el modelo que mejor predice las muertes por millón. Siendo efectivos los predictores: primer año de vacunación, último año de vacunación y el stringency_index, que indica el nivel de las restricciones adoptadas.

Computemos el AIC para comprobar:

```
step(modelo)
```

```
## Start:  AIC=419.85
```

```
## dpm_100d ~ periods_with_vaccine + vaccination_2020_2015 + last_vaccine_year +
##     first_vaccine_year + si_100d
##
##                          Df Sum of Sq    RSS    AIC
## - last_vaccine_year       1      0.90 116496 417.85
## - periods_with_vaccine    1      1.67 116497 417.85
## - first_vaccine_year      1     23.91 116519 417.86
## - si_100d                 1    318.23 116814 418.00
## - vaccination_2020_2015   1    675.55 117171 418.16
## <none>                                116495 419.85
##
## Step:  AIC=417.85
## dpm_100d ~ periods_with_vaccine + vaccination_2020_2015 + first_vaccine_year +
##     si_100d
##
##                          Df Sum of Sq    RSS    AIC
## - periods_with_vaccine    1      4.27 116501 415.85
## - first_vaccine_year      1     26.27 116523 415.86
## - si_100d                 1    320.26 116817 416.00
## - vaccination_2020_2015   1    680.23 117177 416.16
## <none>                                116496 417.85
##
## Step:  AIC=415.85
## dpm_100d ~ vaccination_2020_2015 + first_vaccine_year + si_100d
##
##                          Df Sum of Sq    RSS    AIC
## - first_vaccine_year      1     39.44 116540 413.87
## - si_100d                 1    322.68 116823 414.00
## - vaccination_2020_2015   1   2478.46 118979 414.97
## <none>                                116501 415.85
##
## Step:  AIC=413.87
## dpm_100d ~ vaccination_2020_2015 + si_100d
##
##                          Df Sum of Sq    RSS    AIC
## - si_100d                 1    292.38 116832 412.00
## - vaccination_2020_2015   1   2770.84 119311 413.12
## <none>                                116540 413.87
##
## Step:  AIC=412
## dpm_100d ~ vaccination_2020_2015
##
##                          Df Sum of Sq    RSS    AIC
## - vaccination_2020_2015   1    3416.9 120249 411.53
## <none>                                116832 412.00
##
## Step:  AIC=411.53
## dpm_100d ~ 1
##
## Call:
## lm(formula = dpm_100d ~ 1, data = df)
##
## Coefficients:
```

```
## (Intercept)
##       77.89
```

Con esto confirmamos lo dicho, el modelo dpm_100d ~ last_vaccine_year + first_vaccine_year + si_100d es el mejor.

```
modelo<- lm(formula = dpm_100d ~ last_vaccine_year + first_vaccine_year +
    si_100d, data = df)

summary(modelo)
```

```
##
## Call:
## lm(formula = dpm_100d ~ last_vaccine_year + first_vaccine_year +
##     si_100d, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -78.178 -45.764   2.486  36.096  87.599
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         7.105e+02  1.408e+03   0.504    0.616
## last_vaccine_year  -3.043e-01  4.121e-01  -0.738    0.464
## first_vaccine_year -7.834e-03  4.158e-01  -0.019    0.985
## si_100d            -1.323e-01  3.038e-01  -0.436    0.665
##
## Residual standard error: 49 on 49 degrees of freedom
## Multiple R-squared:  0.02145,    Adjusted R-squared:  -0.03846
## F-statistic: 0.358 on 3 and 49 DF,  p-value: 0.7835
```

Es interesante observar que la relación entre el nivel de restricciones y el número de muertes es positiva. Debido a que el Estimate de esta variable es positivo. Esto indicaría que unas mayores restricciones predicen un mayor número de muertes. Basta con conocer el caso para darse cuenta de que la relación es al contrario: un mayor número de muertes indica un momento peor en la epidemia y una escalada de las restricciones.

### ANOVA

Pasamos a realizar un test anova entre las variables muertes por millón y vacunación entre 2015 y 2020. Lo mismo con las variables muertes por millón y último año de vacunación.

```
aggregate(dpm_100d ~ last_vaccine_year, data = df, mean)
```

```
##   last_vaccine_year  dpm_100d
## 1              1950  92.66667
## 2              1970  89.00000
## 3              1980 120.50000
## 4              1990  24.00000
## 5              2005 119.33333
## 6              2010 104.33333
## 7              2015  33.00000
## 8              2020  72.84211
```

```
aggregate(dpm_100d ~ vaccination_2020_2015, data = df, mean)
```

```
##   vaccination_2020_2015 dpm_100d
## 1                     0 90.66667
```

```
## 2                       1 72.84211
```

Observamos que la diferencia en las medias de muertes por millón en los países con distinta vacunación entre 2015 y 2020 es alta. También observamos diferencias, aunque menos claras, dependiendo del último año de vacunación.

Pasaremos, ahora, a comprobar la normalidad de la variable.

```
by(
   df,
   df %>% .$vaccination_2020_2015,
   FUN=function(x){nortest::lillie.test(x$dpm_100d)}
   )
```

```
## df %>% .$vaccination_2020_2015: 0
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  x$dpm_100d
## D = 0.14297, p-value = 0.5621
##
## ------------------------------------------------------------
## df %>% .$vaccination_2020_2015: 1
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  x$dpm_100d
## D = 0.10966, p-value = 0.2965
```

El test nos indica que podemos asumir una distribución normal de la variable en los países sin vacuna entre 2015 y 2020 pero no en los países con vacuna en estos años, ya que el resultado del test ha sido un p-value por debajo de 0.05.

Comprobaremos lo mismo para la variable last_vaccine_year.

```
tryCatch(
    {
     by(
        df,
        df %>% .$last_vaccine_year,
        FUN = function(x) {nortest::lillie.test(x$dpm_100d)}
        )
     },
    error = function(e) {
      message("Error in nortest::lillie.test(x$dpm_100d) : sample size must be greater than 4")
      }
)
```

```
## Error in nortest::lillie.test(x$dpm_100d) : sample size must be greater than 4
```

En este caso, la comparación entre grupos es imposible ya que no todos los grupos incluyen más de 4 países. Dado que la muestra no cumple el requisito de normalidad, no es adecuado comparar las medias utilizando un test ANOVA.

### Clustering

Pasamos al clustering.

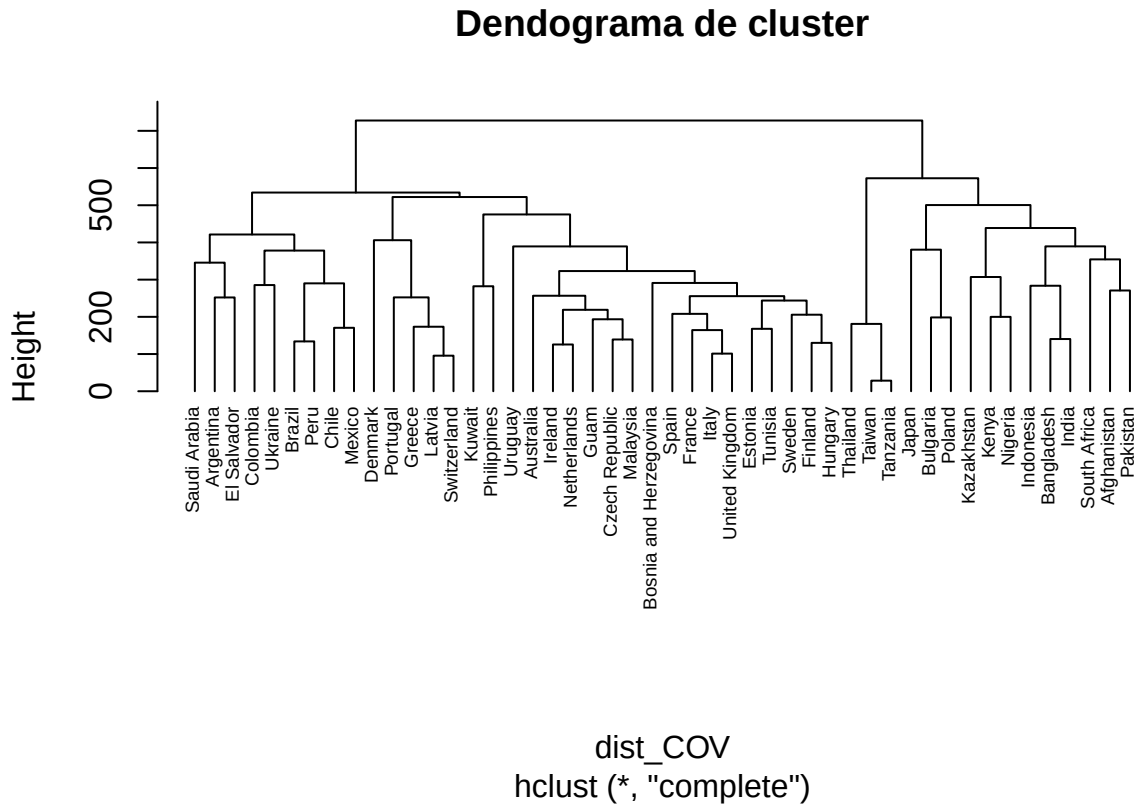
Dado que el clustering se basa en la vectorización de las observaciones y calcular la distancia entre estas,

decidimos utilizar todos los datos disponibles para agrupar los datos. Utilizamos, por lo tanto, un nuevo dataframe que incluye los datos de vacunación lustro a lustro, de muertes en intervalos de tiempo y de stringency index en intervalos de tiempo.

```
COVID_BGC2 <- inner_join(BCG_no_strain_no_NA, COVID_Na_df) %>% na.omit()
```

```
## Joining, by = "country_name"
```

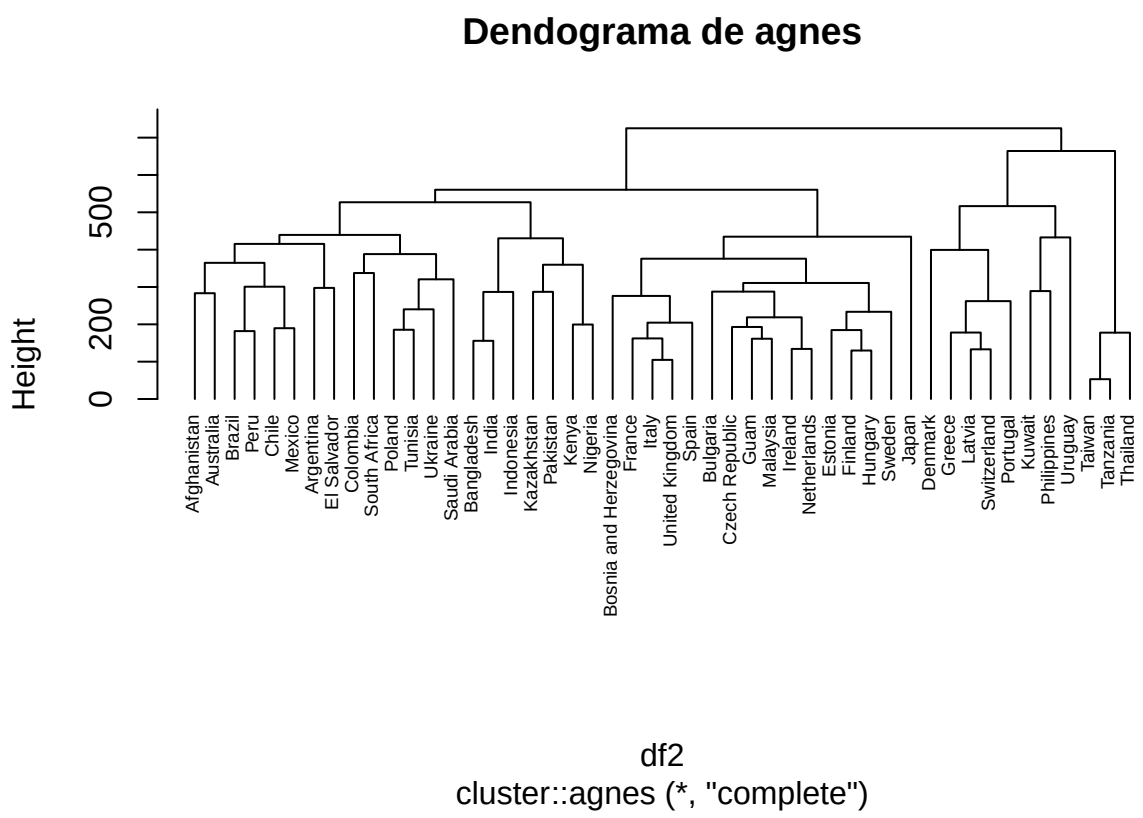
```
df2 <- COVID_BGC2[2:ncol(COVID_BGC2)]
dist_COV <- dist(df2, "euclidean") # Calculamos la distancia euclidea
hc_COV <- hclust(dist_COV, "complete") # Clasificamos en clusters
# hc_COV$labels <- COVID_BGC2[hc_COV$labels,]$country_name
plot(# Visualizar
     hc_COV,
     labels=COVID_BGC2$country_name,
     cex=0.6,
     hang=-1,
     main ="Dendograma de cluster"
     )
```




Computando la distancia euclidea vemos que existen dos grandes clusters, uno con países mucho más parecidos, que presentan menos muertes y una vacunación similar. Y otro en el que encontramos más variabilidad y los países más golpeados por la pandemia.

```
hc_ag_COV <- cluster::agnes(df2,method = "complete") # Clasificamos con agnes
cluster::pltree(# Visualizar
                hc_ag_COV,
                cex=0.6,
```

```
                hang=-1,
                main="Dendograma de agnes",
                labels=COVID_BGC2$country_name
                )
```

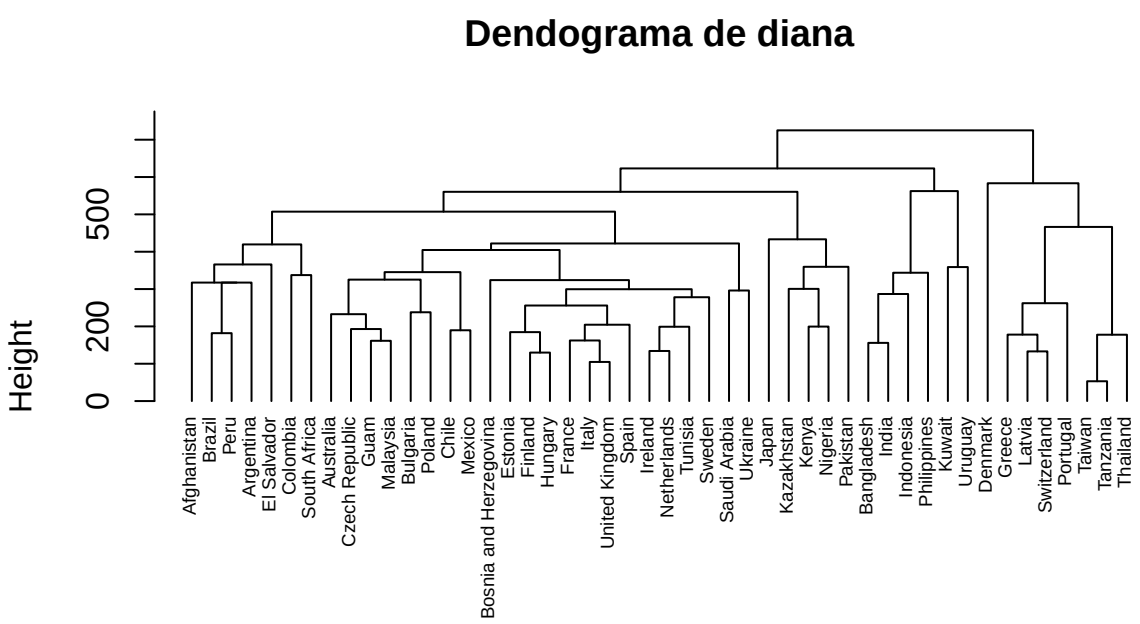



Muy similares son los datos de agnes.

```
print(paste("el coeficiente de aglomeración de agnes es", hc_ag_COV$ac))
```

```
## [1] "el coeficiente de aglomeración de agnes es 0.697709689722129"
```

```
hc_di_COV <- cluster::diana(df2) # Clasificamos por división
cluster::pltree(# Visualizar
                hc_di_COV,
                cex=0.6,
                hang=-1,
                main="Dendograma de diana",
                labels=COVID_BGC2$country_name
                )
```

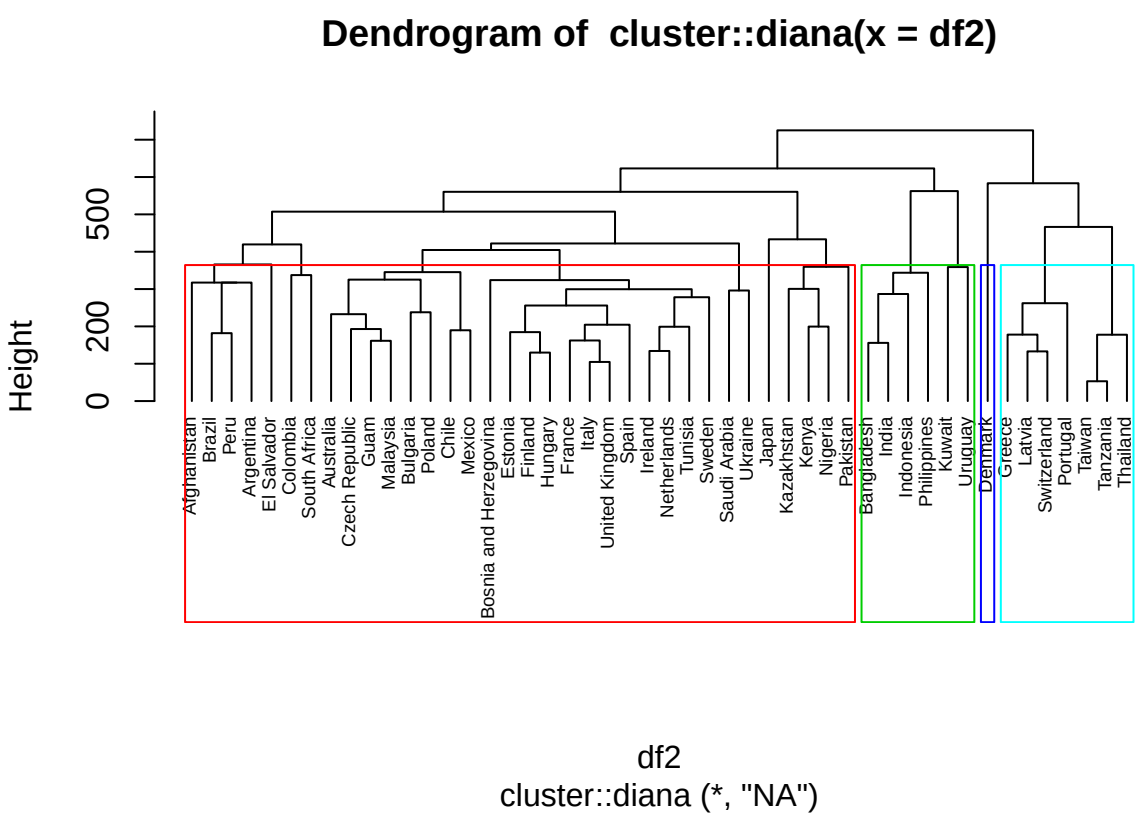



Y muy similares los de diana. En todos ellos, España tiene una gran similaridad con Italia, Suecia y Reino Unido. Los paises europeos con mayor número de muertes.

```
print(paste("el coeficiente de division de diana es", hc_di_COV$dc))
```

```
## [1] "el coeficiente de division de diana es 0.682464805212571"
```

```
#Visualizamos los clusters por colores.
cluster::pltree(hc_di_COV,hang=-1,cex=0.6, labels = COVID_BGC2$country_name)
rect.hclust(hc_di_COV,k=4,border=2:10)
```




Si observamos los cuatro clusters principales en diana vemos que una mayoría de países con una baja incidencia durante la primera ola de la pandemia se encuentran en el cluster rojo, a poca distancia. Luego vemos un cluster compartido por Colombia, Portugal, Suiza y México; y otros dos clusters más. Es posible asumir que estas distancias, por cómo han sido computadas, se deben a diferencias en el aumento de las muertes por coronavirus, en las restricciones adoptadas y en la vacunación de BCG.

## Conclusiones del estudio

Hemos podido ver que existen correlaciones negativas entre las campañas de vacunación de BCG y las muertes por millón debidas a la COVID19. De hecho, hemos podido comprobar que las variables `last_vaccine_year` junto a `first_vaccine_year` sirven como buenos predictores para las muertes por millón (variable `dpm_100d`). Se podría debatir si la variable `si_100d` contribuye a una mejor predicción, ya que su efecto puede ser tanto a priori (menos muertes debido a unas restricciones más severas, hecho que disminuye los contagios) como a posteriori (dado que el país ha sufrido de un gran número de muertes, la severidad de las restricciones se ha visto aumentada para disminuir los contagios).

Con respecto a los clusters de países, como ya se ha mencionado antes, estos se agrupan bien según las severidad de las restricciones y las campañas de vacunación de la BCG.

Como corolario de los resultados anteriores podemos decir que a priori la vacuna BCG tiene un efecto beneficioso para el sistema inmune, dotando a la población de una mayor supervivencia frente a la COVID19. Estos resultados deberían respaldarse con estudios de comparación de secuencias de los antígenos proporcionados por la vacuna BCG y el genoma del SARS-CoV-2 ya que en caso de hallar similitudes, estas podrían explicar el fenómeno observado. Podría ser que los anticuerpos generados como respuesta a la vacuna BCG tuvieran afinidad por antígenos presentados por el SARS-CoV-2 y por tanto activarían una primera respuesta inmune ante la infección. Estos resultados podrían emplearse también para acelerar el proceso de desarrollo de vacunas para la COVID19.