

# Análisis de la evolución de la incidencia de la COVID-19 en España desde el 1 de enero de 2020 hasta la actualidad

Una exploración de los datos de casos detectados de la COVID-19 y la mortalidad en España durante la pandemia mediante R y R Markdown (RStudio ®)

Juan Matorras Díaz-Caneja

23/02/2021

## 1.- Introducción

Este es un ejercicio básico de análisis de los datos de la incidencia de la COVID-19 en España desde el comienzo del año 2020. Habiendo la cantidad de informes y herramientas para el análisis de los datos sobre la incidencia de la COVID-19 que ya existen, este documento no pretende aportar nada singularmente nuevo y su razón de ser no es otra que poner en práctica y profundizar por mi parte en el aprendizaje de las técnicas de análisis de datos y el lenguaje R que inicié en la segunda mitad de septiembre de 2020.

Hay que matizar que, si bien este estudio empezó cubriendo únicamente fechas dentro de 2020, por la propia duración de la pandemia, el alcance del mismo se ha extendido para acomodar los datos disponibles correspondientes ya al año 2021.

Por su propia esencia, éste es un documento vivo que además de ser puesto al día periódicamente con los nuevos datos disponibles, va sufriendo adiciones, modificaciones y correcciones de erratas. La última versión disponible de este documento y de los datos empleados en su elaboración se pueden encontrar en el repositorio de GitHub: <https://github.com/JuanMatorras/Covid-19>.

Los datos de partida son los publicados por el Gobierno de España. Estos datos estaban durante 2020 disponibles directamente a través la web **datos.gob.es** en el enlace: <https://datos.gob.es/es/catalogo/e05070101-evolucion-de-enfermedad-por-el-coronavirus-covid-19>. Con el cambio de año este enlace ha quedado anulado y es necesario acudir a: <https://cnecovid.isciii.es/covid19/#documentación-y-datos> donde se encuentran los nuevos enlaces de descarga. En este caso el archivo utilizado es el de datos por CCAA: [https://cnecovid.isciii.es/covid19/resources/casos\\_tecnica\\_ccaa.csv](https://cnecovid.isciii.es/covid19/resources/casos_tecnica_ccaa.csv)

El grueso del informe se centra sobre los totales en España agregando los datos disponibles por Comunidades Autónomas, aunque también se muestran información de la incidencia en las CCAA de Madrid, Cantabria, Asturias, Galicia y Castilla y León. La razón de la selección de estas comunidades y no otras responde a consideraciones personales y no obedece a ningún criterio técnico.

## 2.- Proceso metodológico y software utilizado

```
## Funciones y parámetros globales
ggplotCovid <- function(...){ggplot(...) + theme_bw(base_size=22)}
poblESP <- 47.332E+06
```

El archivo de datos no ha sido sometido a ningún tipo de modificación o alteración previa y su manipulación en este análisis es el mínimo imprescindible para permitir el tratamiento de los datos y obtención de resultados.

Al ejecutar el código se descargan los archivos de datos directamente de la web si no se encuentran ya disponibles en el directorio /data.

```

## Creación de directorio para datos origen "data" si no existe ya
if(!file.exists("./data")) {dir.create("./data")}
## Direcciones URL de descarga de los datos de origen
urlCasos <- "https://cnecovid.isciii.es/covid19/resources/casos_tecnica_ccaa.csv"
urlMoMo <- "https://momo.isciii.es/public/momo/data"
## Descarga de archivos
## ***** ¡¡¡¡¡ AVISO IMPORTANTE !!!!! *****
## Si hay archivos preexistentes no se lleva a cabo la descarga.
## Para descargar nuevos archivos desde la web borre las versiones previas
## o trasládelas a otro directorio si quiere conservar un histórico de archivos.
if(!file.exists(file.path("data", "casos_tecnica_ccaa.csv"))) {
  download.file(urlCasos, destfile = file.path("data", "casos_tecnica_ccaa.csv"),
    method = "curl")
}
if(!file.exists(file.path("data", "data.csv"))) {
  download.file(urlMoMo, destfile = file.path("data", "data.csv"), method = "curl")
}

## Lectura de datos de origen
DatosCompleto <- read.csv(file.path("data", "casos_tecnica_ccaa.csv"))
## Descartamos el método de detección (PCR, antígenos,...)
DatosCCAA <- DatosCompleto[, 1:3]
## Adecuación formato fechas para su correcta lectura
DatosCCAA$fecha <- as.Date(DatosCCAA$fecha, format = "%Y-%m-%d")
## Agrupación de información de CCAA para datos a nivel nacional
Datos <- DatosCCAA %>% group_by(fecha) %>% summarise(num_casos = sum(num_casos),
  .groups = 'drop')

```

La fecha y hora de descarga de los datos que han sido utilizados para la elaboración de las tablas y gráficos incluidos en este informe ha sido (aaaa-mm-dd hh:mm:ss): **2021-02-23 10:01:42**

El análisis se ha llevado a cabo utilizando el entorno de desarrollo integrado de **RStudio** © versión 1.4.1103 (1) para el software libre de análisis estadístico **R**, versión 4.0.3 (2).

Se ha hecho uso también de los siguiente paquetes complementarios para R:

- **lubridate** ver.1.7.9.2 para facilitar el manejo de fechas. (3)
- **knitr** ver.1.31 para mejorar la apariencia de tablas. (4)
- **tidyverse** ver.1.3.0 por los paquetes que incluye para ayudar en la extracción y manipulación de la información en tablas y los gráficos mejorados de **ggplot2** ver.3.3.3. (5) y (6)
- **data.table** ver.1.13.6 con el objeto de manipular las tablas más eficientemente. (7)

Nota 1: La codificación empleada en los datos fuente para la identificación de comunidades autónomas y provincias se corresponde con la ISO 3166-2:ES. La correspondencia entre dicha codificación y la comunidad autónoma o provincia referida puede ser consultada, entre otras fuentes, en: [https://es.wikipedia.org/wiki/ISO\\_3166-2:ES](https://es.wikipedia.org/wiki/ISO_3166-2:ES)

Nota 2: En algunos puntos la programación puede ser lejos de óptima, en primera instancia por mi falta de fluidez en la programación con R por estar en proceso de aprendizaje, pero también afectado significativamente por la manera en la que se ha ido construyendo este documento, que empezó siendo algo sencillo, muy limitado en su alcance, pero que ha sufrido múltiples ampliaciones en su alcance en sucesivas etapas y nunca siguiendo un guión definido. Un ejemplo claro son los apartados donde se compara la información entre CCAA. En primera instancia empezaron siendo sólo dos comunidades y por eso se programó de una manera, que se debería haber modificado cuando la lista empezó a hacerse larga a medida que incorporaba nuevas CCAA a los análisis comparativos.

Nota 3: Hay que llamar la atención sobre la naturaleza de los datos, los cuales proceden de la agregación de las

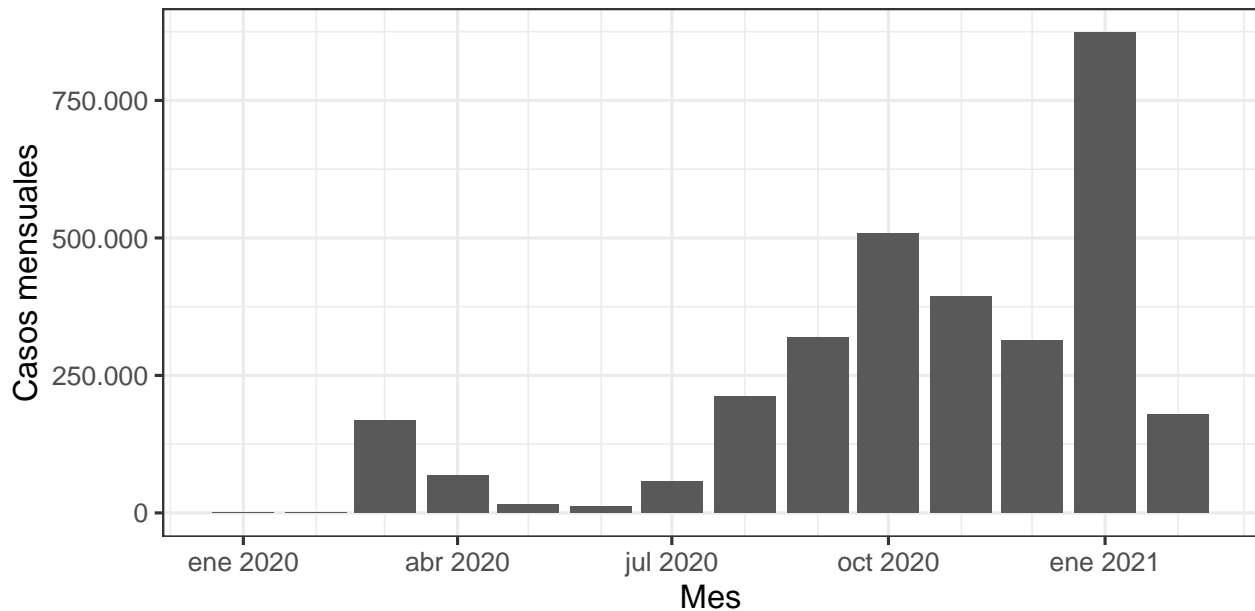
aportaciones desde las CCAA al sistema RENAVE. Esto resulta en que los datos de casos de fechas recientes no están completos y esto hace que todas las gráficas con desagregación diaria o variables derivadas, como la incidencia acumulada, presenten siempre un tramo descendente en su extremo final, con independencia de si realmente el progreso de la enfermedad es de expansión o de contracción.

### 3.- Casos por meses y número total de casos detectados desde el inicio de 2020

La evolución de número de casos notificados por meses se refleja en el gráfico que se muestra a continuación:

```
dfCasosMensuales <- Datos %>% group_by(Mes = floor_date(fecha, "month")) %>%
  summarize(Casos = sum(num_casos), .groups = "drop")
TablaCasosMensuales <- data.frame("Mes" = format(dfCasosMensuales$Mes, "%b %Y"),
  "Casos" = format(dfCasosMensuales$Casos,
    big.mark = ".",
    decimal.mark = ","))

TotalCasosOrigen <- sum(Datos$num_casos)
PorcentajePoblacion <- paste(format(TotalCasosOrigen / poblESP *100, digits = 4,
  decimal.mark = ","), "%")
d <- ggplotCovid(dfCasosMensuales, aes(Mes, Casos))
d <- d + labs(y = "Casos mensuales")
d <- d + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
  big.mark=".",
  decimal.mark=",")})
d <- d + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(dfCasosMensuales$Mes,1),
  by="3 months"),
  date_minor_breaks = "1 month",
  date_labels = "%b %Y")
d + geom_col()
```



Correspondiente a los valores que se incluyen en la tabla siguiente:

Mes	Casos
ene 2020	45
feb 2020	1.728
mar 2020	167.358

Mes	Casos
abr 2020	68.311
may 2020	16.100
jun 2020	11.134
jul 2020	57.608
ago 2020	212.231
sep 2020	318.939
oct 2020	508.200
nov 2020	394.013
dic 2020	313.960
ene 2021	873.169
feb 2021	178.907

El número total de casos acumulados desde el 1 de enero de 2020 hasta la fecha indicada en el punto anterior según los datos oficiales disponibles en ese momento ascienden a un total de **3.121.703** personas.

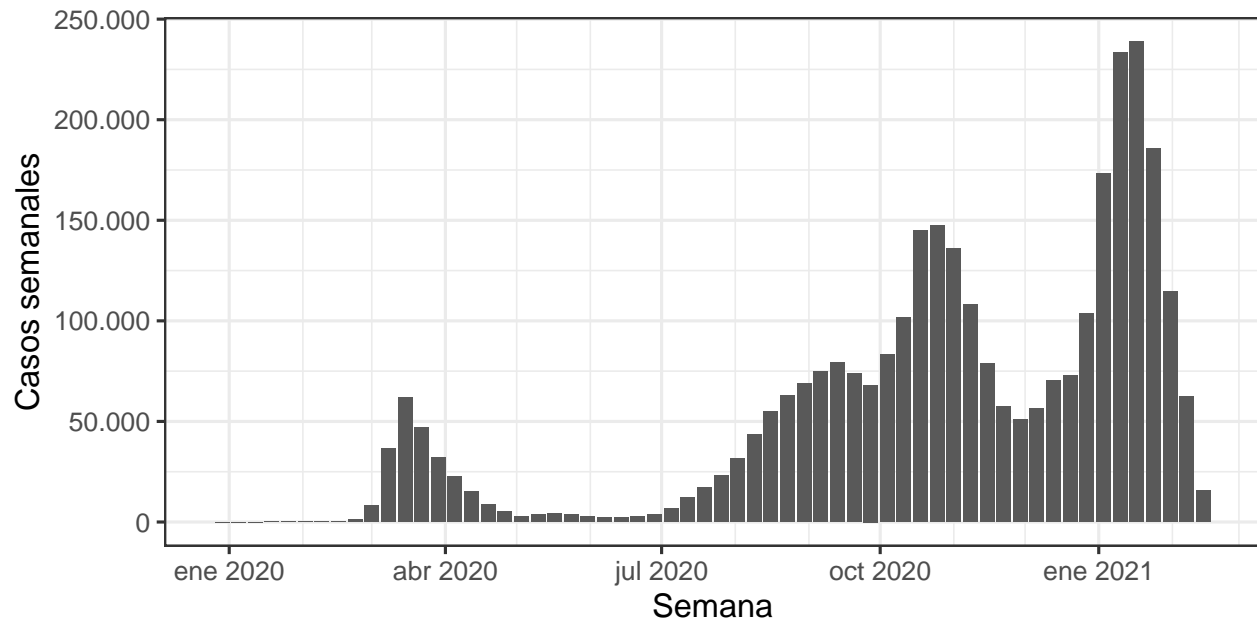
Lo que en cualquier caso es de conocimiento general es la carestía de pruebas diagnósticas durante la primera fase de la pandemia, lo que apunta a una clara subestimación de los casos reales en esos primeros meses de 2020, particularmente en marzo y abril. Cabría intentar estimar el número real de casos en estos primeros meses aplicando el ratio de pruebas con resultado positivo respecto a pruebas realizadas en la segunda mitad de año al número de pruebas realizadas en dichos meses iniciales pero no deja de ser una aproximación un tanto burda y harían falta datos fiables del número de pruebas que se han realizado, tanto en el ámbito público como privado, además de necesitarse datos complementarios ya que los casos detectados luego tienen múltiples pruebas de seguimiento.

Considerando una población en España de **47,332** millones de personas según los datos publicados por el INE (Instituto Nacional de Estadística) correspondientes al inicio del año 2020, el porcentaje de contagio de la población es del **6,595 %** hasta la fecha. Insistimos en que, puesto que la incidencia de la enfermedad en los primeros meses de 2020 está por fuerza minusvalorada por la escasez de pruebas de diagnóstico, y no pudiendo olvidar que además tenemos el fenómeno de los casos de infección asintomáticos, el porcentaje de población afectada realmente es necesariamente más alto que el reflejado en este cálculo y es más que probable que sea del orden del doble. De hecho, para poder cubrir esta laguna y tener una idea más fiable del verdadero impacto de la enfermedad sobre el total de la población están los estudios de sero-epidemiología que se han venido realizando desde junio de 2020 (<https://portalcne.isciii.es/enecovid19/>).

## 4.- Casos semanales

Evolución de número de casos identificados por semanas:

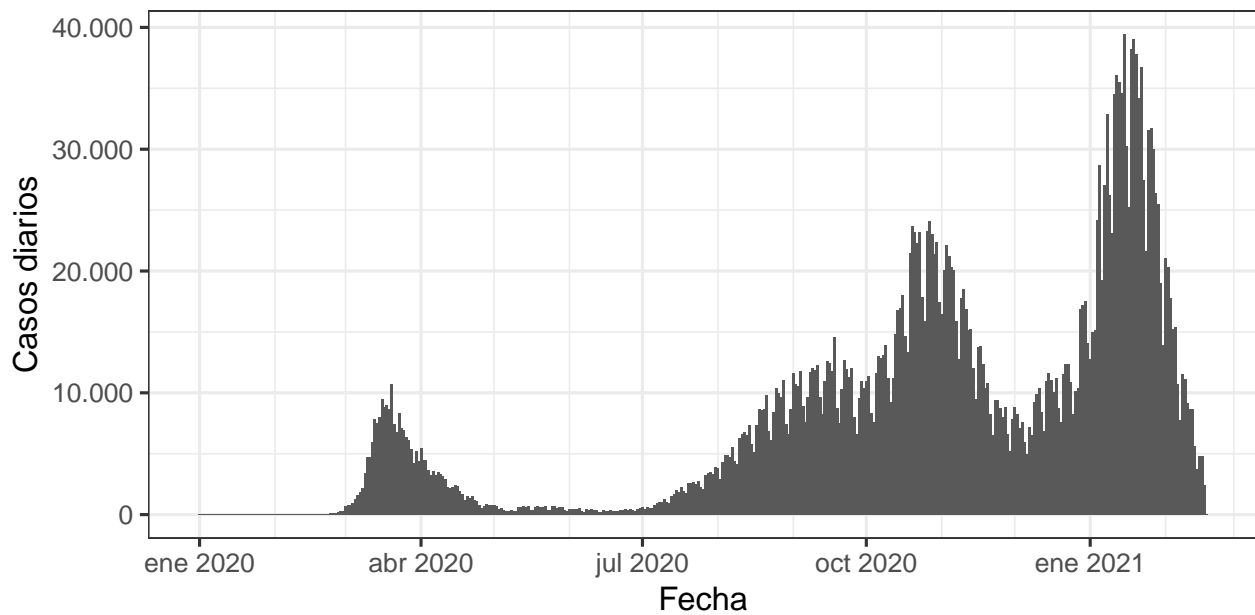
```
dfCasosSemanales <- Datos %>% group_by(Semana = floor_date(fecha, "week")) %>%
  summarize(Casos = sum(num_casos), .groups = "drop")
d <- ggplotCovid(dfCasosSemanales, aes(Semana, Casos))
d <- d + labs(y = "Casos semanales")
d <- d + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
  big.mark=".",
  decimal.mark=",")})
d <- d + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(dfCasosSemanales$Semana,1),
  by="3 months"),
  date_minor_breaks = "1 month",
  date_labels = "%b %Y")
d + geom_col()
```



## 5.- Casos diarios

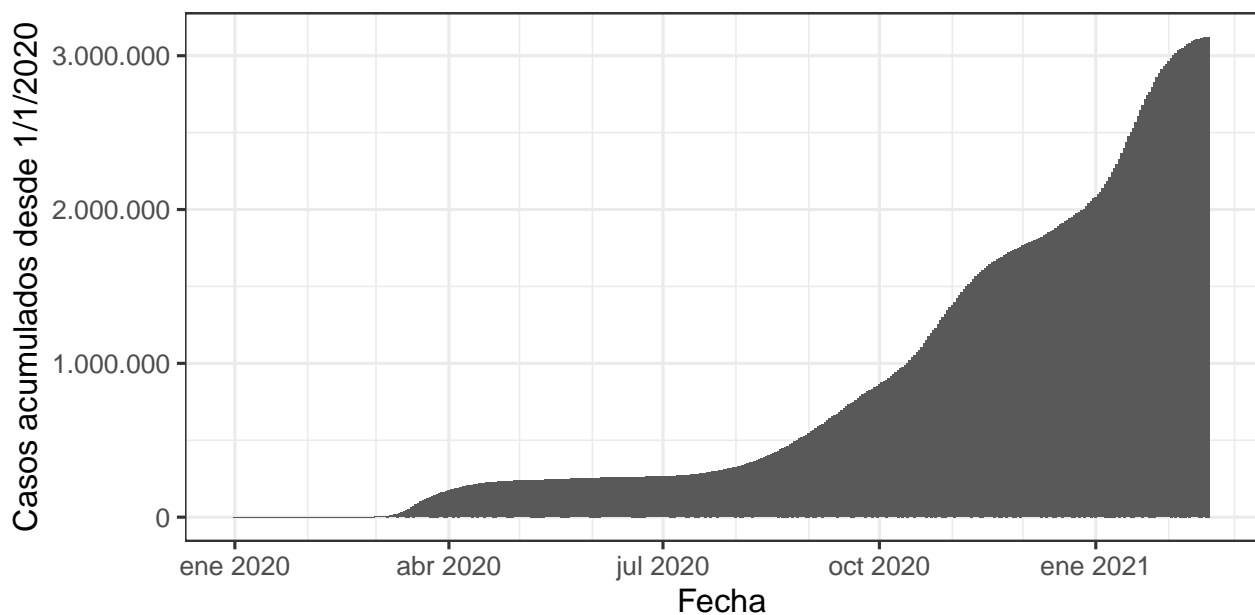
- Curva epidémica de los casos notificados por días:

```
dfCasosDiarios <- data.frame(Fecha = Datos$fecha, Casos = Datos$num_casos)
d <- ggplotCovid(dfCasosDiarios, aes(Fecha, Casos))
d <- d + labs(y = "Casos diarios")
d <- d + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                    big.mark=".",
                                                    decimal.mark=",")})
d <- d + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(dfCasosDiarios$Fecha,1),
                                by="3 months"),
                    date_minor_breaks = "1 month",
                    date_labels = "%b %Y")
d + geom_col()
```



- Gráfico de casos acumulados a origen por días:

```
CasosDiariosAcumul <- cumsum(dfCasosDiarios$Casos)
dfCasosDiariosAcumul <- data.frame(Fecha = Datos$fecha, Casos = CasosDiariosAcumul)
d <- ggplotCovid(dfCasosDiariosAcumul, aes(Fecha, Casos))
d <- d + labs(y = "Casos acumulados desde 1/1/2020")
d <- d + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                    big.mark=".",
                                                    decimal.mark=",")})
d <- d + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(dfCasosDiariosAcumul$Fecha,1),
                                by="3 months"),
                    date_minor_breaks = "1 month",
                    date_labels = "%b %Y")
d + geom_col()
```

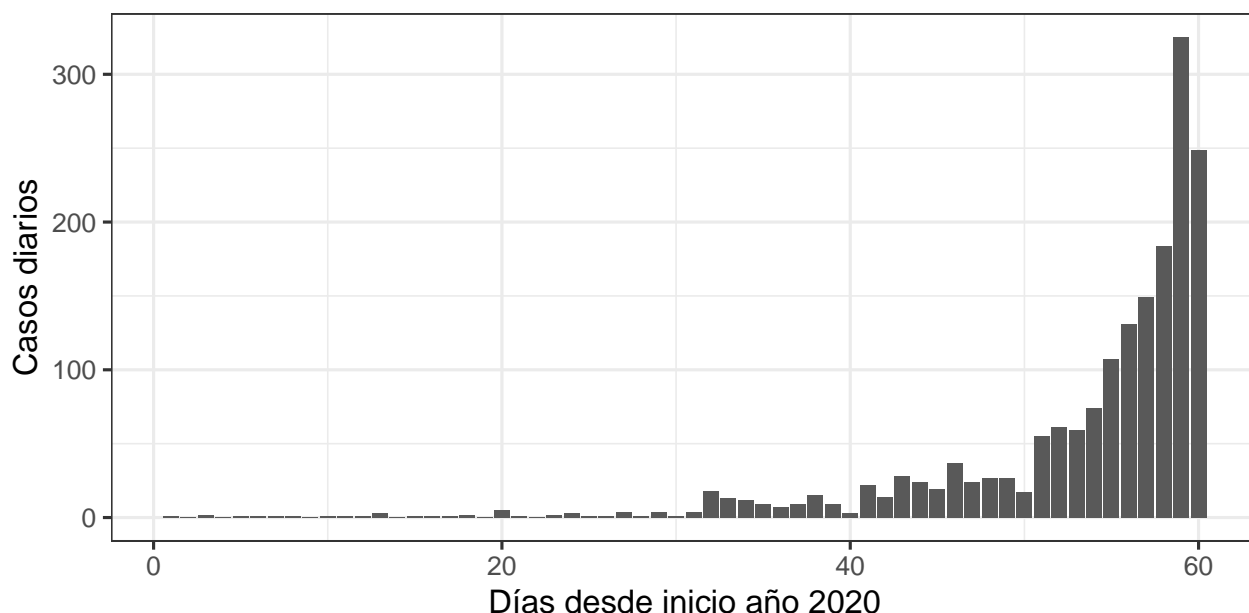


El resultado de esta desagregación de datos, o más estrictamente de la ausencia de agregación de los mismos, puesto que los datos de partida son diarios, es la aparición de dientes de sierra que son simplemente el reflejo del descenso en el número de pruebas realizadas en fines de semana y festivos en comparación con los realizados en días laborables.

## 6.- Detalle del número de casos en los primeros meses de 2020

- Evolución diaria del número de casos durante los dos primeros meses del año:

```
PeriodoEF <- seq.Date(
  from = as.Date("2020-01-01"),
  to = as.Date("2020-02-29"),
  by = "day")
DatosEF <- subset(Datos, fecha %in% PeriodoEF)
CasosDiariosEF <- tapply(DatosEF$num_casos, yday(DatosEF$fecha), sum)
dfCasosDiariosEF <- data.frame(Dia = 1:length(CasosDiariosEF), Casos = CasosDiariosEF)
TotalCasosEF <- sum(DatosEF$num_casos)
d <- ggplotCovid(dfCasosDiariosEF, aes(Dia, Casos))
d <- d + labs(x = "Días desde inicio año 2020")
d <- d + labs(y = "Casos diarios")
d <- d + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
  big.mark=".",
  decimal.mark=",")})
d + geom_col()
```



**NOTA IMPORTANTE:** Este gráfico ha permitido detectar un problema en los datos de origen que se manifestó en los primeros días de 2021, apareciendo casos asignados a los primeros días de 2020 que antes no existían. Esto se debió a un deficiente registro de nuevos casos correspondientes realmente a 2021 que se están dando de alta con fecha de 2020. En el repositorio de GitHub al que se hace referencia al principio del documento se adjunta una tabla comparativa donde se puede observar cómo fueron evolucionando el número de casos declarados para enero de 2020 en las distintas CCAA entre los días finales de diciembre de 2020 y primeros días de 2021. Las mayores distorsiones se aprecian en los datos de las comunidades autónomas de Madrid y Cataluña en las que para los primeros días de enero de 2020 tenían declarados números de casos con cifras muy bajas, en general 0 o 1, y que durante los primeros días de enero de 2021 empiezan a mostrar cantidades de casos que superan la veintena y la treintena para esas fechas del año 2020. Se puso en

conocimiento de este problema al Centro de Estudios Epidemiológicos del Instituto de Salud Carlos III el día 8 de enero de 2021 desde donde confirmaron el problema el día 19 de enero, informando que se esperaba corregir el problema en los días siguientes. Finalmente, a partir de la segunda semana de febrero los errores parecen haber sido ya corregidos.

Los casos reportados totales a lo largo de esos dos meses son **1.773**, si bien es claro que se produce una acusada inflexión en la pendiente de crecimiento a partir del día 50.

```
Periodo1_50 <- seq.Date(  
  from = as.Date("2020-01-01"),  
  to = as.Date("2020-02-19"),  
  by = "day")  
Periodo51_60 <- seq.Date(  
  from = as.Date("2020-02-20"),  
  to = as.Date("2020-02-29"),  
  by = "day")  
Datos1_50 <- subset(Datos, fecha %in% Periodo1_50)  
Datos51_60 <- subset(Datos, fecha %in% Periodo51_60)  
TotalCasos1_50 <- sum(Datos1_50$num_casos)  
TotalCasos51_60 <- sum(Datos51_60$num_casos)
```

Siendo así que el desglose de número agregado de casos identificados en dichos primeros 50 días y los siguientes 10 días queda de la siguiente manera:

- Periodo 1-50: 379
- Periodo 51-60: 1.394

En 10 días se detectan **3,7** veces los casos que se habían producido en los 50 días anteriores.

#### - Subsiguiente evolución durante la primera quincena de marzo:

En este apartado analizamos cómo continúa desarrollándose la propagación de la pandemia a principios del mes de marzo, estableciendo por su relevancia en lo ocurrido en España durante esos días dos periodos de tiempo diferenciados, del 1 al 8 y del 9 al 15.

```
Periodo1_8mar <- seq.Date(  
  from = as.Date("2020-03-01"),  
  to = as.Date("2020-03-08"),  
  by = "day")  
Periodo9_15mar <- seq.Date(  
  from = as.Date("2020-03-09"),  
  to = as.Date("2020-03-15"),  
  by = "day")  
Datos1_8mar <- subset(Datos, fecha %in% Periodo1_8mar)  
Datos9_15mar <- subset(Datos, fecha %in% Periodo9_15mar)  
CasosDiarios1_8mar <- tapply(Datos1_8mar$num_casos, yday(Datos1_8mar$fecha), sum)  
CasosDiarios9_15mar <- tapply(Datos9_15mar$num_casos, yday(Datos9_15mar$fecha), sum)  
TotalCasos1_8mar <- sum(Datos1_8mar$num_casos)  
TotalCasos9_15mar <- sum(Datos9_15mar$num_casos)
```

En los primeros ocho días de marzo la progresión diaria de nuevos casos siguió disparándose, resultando un total de **10.235** casos a añadir al total anterior, siendo éstos **5,8** veces los registrados a lo largo de todo enero y febrero.

Durante los siguientes siete días, del 9 al 15 de marzo, los casos a sumar fueron **42.095**, lo que supone **4,1** veces los notificados en los 8 primeros días del mes.

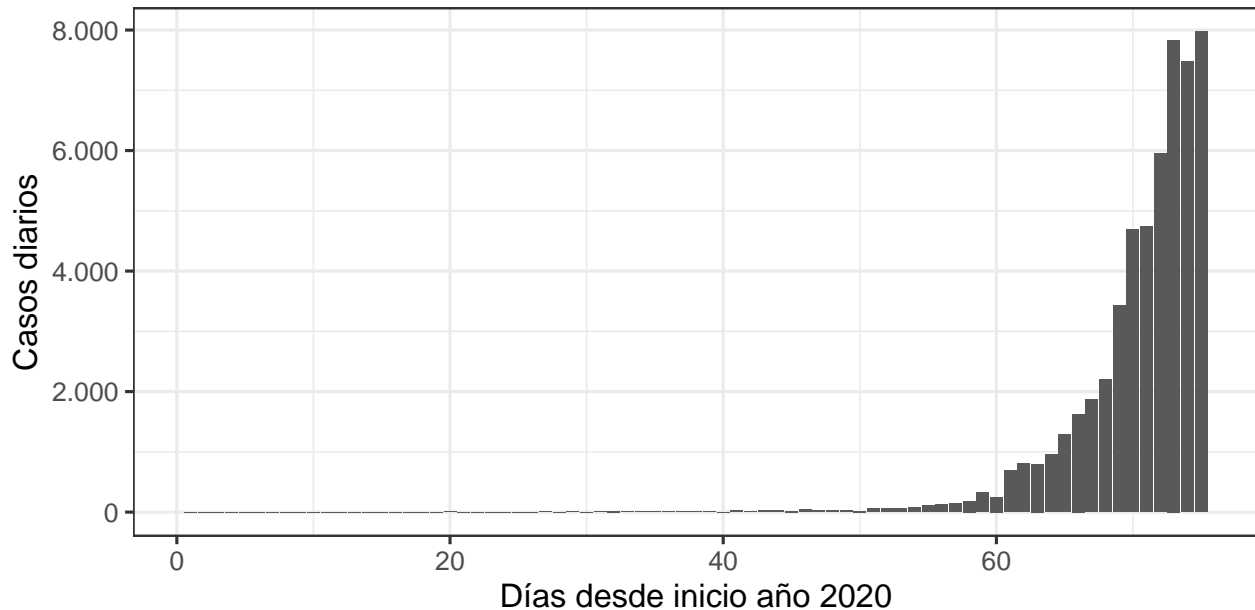


- Gráfico del número de casos diarios desde el 1 de enero hasta el 15 de marzo de 2020:

```
CasosDiariosHasta15mar <- c(CasosDiariosEF, CasosDiarios1_8mar, CasosDiarios9_15mar)
dfCasosDiariosHasta15mar <- data.frame(Fecha = 1:length(CasosDiariosHasta15mar), Casos =
                                     CasosDiariosHasta15mar)

d <- ggplotCovid(dfCasosDiariosHasta15mar, aes(Fecha, Casos))
d <- d + labs(x = "Días desde inicio año 2020")
d <- d + labs(y = "Casos diarios")
d <- d + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                       big.mark=".",
                                                       decimal.mark=",")})

d + geom_col()
```



- Incidencia acumulada por 100.000 habitantes en los 14 días previos a la declaración del estado de alarma del 14 de marzo

Pasemos ahora a calcular la incidencia acumulada por cada 100.000 habitantes en los 14 días previos a la declaración del estado de alarma que tuvo efecto

```
PeriodoPrevioEstadoAlarma <- seq.Date(
  from = as.Date("2020-02-29"),
  to = as.Date("2020-03-13"),
  by = "day")
DatosPrevioEstadoAlarma <- subset(Datos, fecha %in% PeriodoPrevioEstadoAlarma)
IncidAcumPrevioEstadoAlarma <- sum(DatosPrevioEstadoAlarma$num_casos)/poblESP*1E+05
```

Tomando esos 14 días previos, es decir, entre el 29 de febrero y el 13 de marzo, la incidencia acumulada por cada 100.000 habitantes, con el mismo dato de población presentado más arriba fue de **78** casos/100.000 hab.

Contrasta este valor de forma muy llamativa con los límites que se han estado manejando en España durante la segunda ola de infecciones, donde se ha hablado de 200, 500 e incluso 1.000 casos/100.000 hab. para empezar a tomar medidas de limitación de la movilidad y el contacto entre personas, valores totalmente irracionales y desproporcionados en comparación con los manejados por otros países de la Unión Europea.

## 7.- Evolución de la incidencia acumulada a lo largo de todo el periodo de análisis

En el siguiente gráfico se representan las incidencias acumuladas por cada 100.000 habitantes correspondientes a periodos de 14 y 7 días:

```
Fechas <- seq.Date(as.Date("2020-01-01"), by = "day", length.out =
                    length(dfCasosDiarios$Fecha))
UltFecha <- Fechas[length(Fechas)]
stopifnot(UltFecha == max(Datos$fecha)) ## Prueba interna consistencia datos

## Inicialización de variables
InicioPeriodo14 <- as.Date("2020-01-01")
FinPeriodo14 <- InicioPeriodo14 + 13
IncidAcum14 <- data.frame(Fecha=as.Date(character()), IA14=double())
# IA14aux <- as.double()
InicioPeriodo7 <- as.Date("2020-01-01")
FinPeriodo7 <- InicioPeriodo7 + 6
IncidAcum7 <- data.frame(Fecha=as.Date(character()), IA7=double())

## Incidencia acumulada 14 días
while (FinPeriodo14 <= UltFecha) {
  Periodo14 <- seq.Date(
    from = InicioPeriodo14,
    to = FinPeriodo14,
    by = "day")
  DatosIA14 <- subset(Datos, fecha %in% Periodo14)
  IncidAcum14 <- add_row(IncidAcum14, Fecha = FinPeriodo14,
                        IA14 = sum(DatosIA14$num_casos)/poblESP*1E+05)
  InicioPeriodo14 <- InicioPeriodo14 + 1
  FinPeriodo14 <- FinPeriodo14 + 1
}

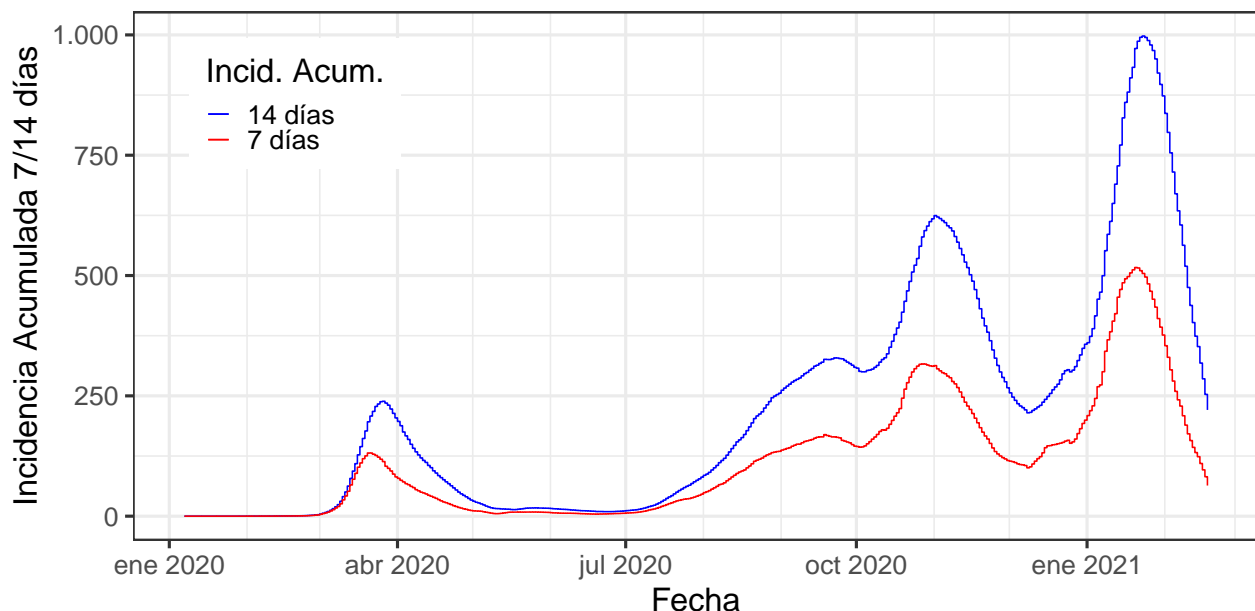
## Incidencia acumulada 7 días
while (FinPeriodo7 <= UltFecha) {
  Periodo7 <- seq.Date(
    from = InicioPeriodo7,
    to = FinPeriodo7,
    by = "day")
  DatosIA7 <- subset(Datos, fecha %in% Periodo7)
  IncidAcum7 <- add_row(IncidAcum7, Fecha = FinPeriodo7,
                       IA7 = sum(DatosIA7$num_casos)/poblESP*1E+05)
  InicioPeriodo7 <- InicioPeriodo7 + 1
  FinPeriodo7 <- FinPeriodo7 + 1
}

## Gráficos de Incidencia Acumulada superpuestos
## Combinación de ambos data frames en IncidAcum7_14
IncidAcum7_14 <- add_column(IncidAcum7, IA14=c(rep(0, 7), IncidAcum14$IA14))
## 7 primeros valores deberían ser NA, 0 para evitar aviso
g <- ggplotCovid(IncidAcum7_14, aes(Fecha))
g <- g + labs(y = "Incidencia Acumulada 7/14 días")
g <- g + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                       big.mark=".",
                                                       decimal.mark=",")})
g <- g + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(IncidAcum7_14$Fecha,1),
```

```

      by="3 months"),
      date_minor_breaks = "1 month",
      date_labels = "%b %Y")
g <- g + scale_color_manual(name = "Incid. Acum.",
                           values = c("14 días" = "blue", "7 días" = "red"))
g <- g + geom_step(aes(y=IA14, color = "14 días"))
g <- g + geom_step(aes(y=IA7, color = "7 días"))
g <- g + theme(legend.position=c(.05, .95), legend.justification=c("left", "top"))
g

```



Detengámonos un momento en este punto para analizar estas curvas, primero comparando la incidencia acumulada por cada 100.000 habitantes en 14 días (IA14) con la acumulada en 7 días (IA7), para luego centrarnos en la propia evolución temporal de estos indicadores.

Dentro de los propios círculos de científicos epidemiológicos existe cierta disparidad de opinión respecto a si hacer el seguimiento de la IA14, la IA7 o incluso ambas en paralelo. En primer lugar no perdamos de vista que el hecho de agregar casos identificados durante un periodo determinado de tiempo, sea éste de 7 días o de 14 días, lo que se persigue con esta operación es laminar los picos que se producen en la detección de casos diarios, principalmente por la influencia de fines de semana y festivos que ya vimos en el punto 5 y así poder determinar con más facilidad la evolución de la propagación de una determinada enfermedad entre la población. La razón para ampliar de 7 a 14 días debería estar en poder absorber mejor las oscilaciones que se provocan por festivos y puentes y no tanto en el mayor o menor periodo de incubación de la enfermedad. De hecho, el efecto laminador de la IA14 frente a la IA7 se puede observar claramente al comparar ambas curvas en el mes de diciembre de 2020. Es más, contrariamente a lo que se pueda presuponer sobre el hecho de que utilizar un periodo más corto de tiempo ayude a tener una mejor idea de la evolución de la enfermedad, esto es solo cierto cuando la curva de incidencia empieza a descender pero no para el momento de aumento de la incidencia. Como se comprueba en la gráfica adjunta, la IA14 empieza a ascender y a mayor ritmo que la IA7, mientras que el descenso de la primera sí que se demora con respecto a la segunda. Lo que sí es lógico y resulta evidente es que los límites de alarma deben ser distintos para cada una de ellas.

Pasando ahora al análisis de la evolución de ambos indicadores y obviando el hecho ya comentado de la subestimación del primer pico de la ola por la baja realización de pruebas de diagnóstico, queda claro que, a pesar de que los medios de comunicación y, en consecuencia, la propia población general hablen de una tercera ola de la pandemia que arranca en el mes de diciembre, no habiendo bajado la IA14 de 200 sería más apropiado hablar de una segunda fase de la segunda ola puesto que ésta no se puede decir que llegó a

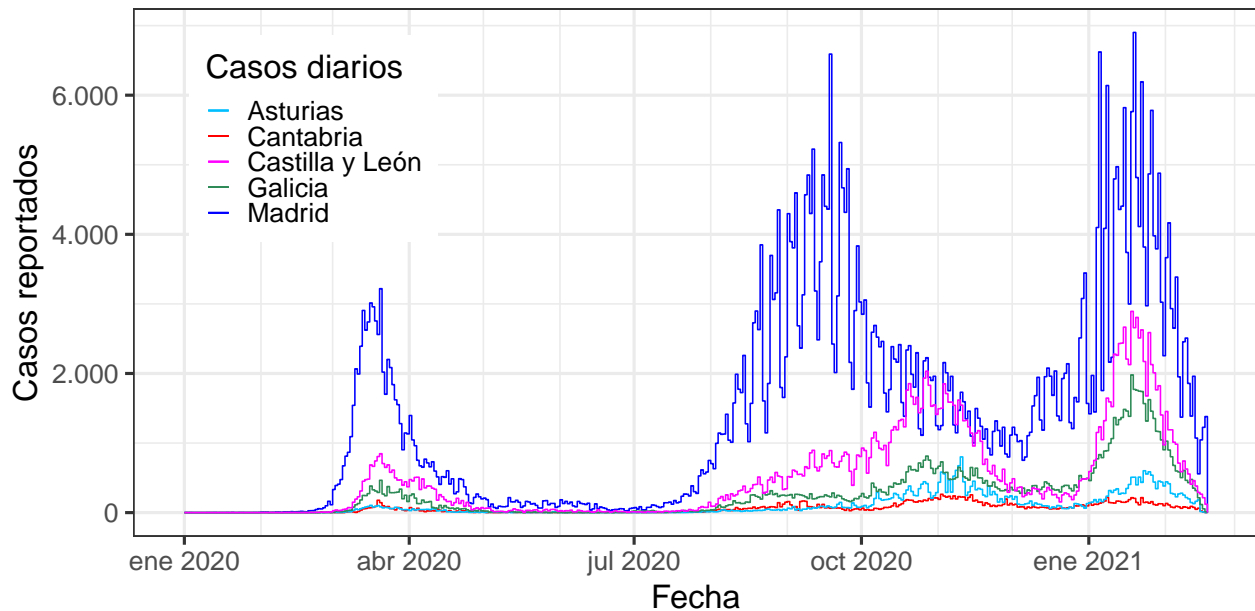
estar bajo control. Si analizamos el impacto de la enfermedad desde el punto de vista de la mortalidad (ver apartado 10) sí que podríamos afirmar de algún modo que la segunda ola habría terminado en diciembre, en tanto en cuanto en este mes se cierra el segundo periodo de exceso de mortalidad, pero no así desde el punto de vista de la incidencia acumulada.

## 8.- Comparación de la evolución del número de casos entre Madrid y otras CCAA

### - Casos reportados por CCAA

En el siguiente gráfico se compara la evolución de la enfermedad entre comunidades muy diferentes, la Comunidad Autónoma de Madrid, Cantabria, Asturias, Galicia y Castilla y León:

```
## ISO CCAA: Madrid= MD, Cantabria= CB, Asturias= AS, Galicia= GA, Castilla y León= CL
CasosDiariosMD <- filter(DatosCCAAs, ccaa_iso == "MD")
CasosDiariosMD <- subset (CasosDiariosMD, select = -ccaa_iso)
CasosDiariosCB <- filter(DatosCCAAs, ccaa_iso == "CB")
CasosDiariosCB <- subset (CasosDiariosCB, select = -ccaa_iso)
CasosDiariosAS <- filter(DatosCCAAs, ccaa_iso == "AS")
CasosDiariosAS <- subset (CasosDiariosAS, select = -ccaa_iso)
CasosDiariosGA <- filter(DatosCCAAs, ccaa_iso == "GA")
CasosDiariosGA <- subset (CasosDiariosGA, select = -ccaa_iso)
CasosDiariosCL <- filter(DatosCCAAs, ccaa_iso == "CL")
CasosDiariosCL <- subset (CasosDiariosCL, select = -ccaa_iso)
CasosDiariosCCAA <- add_column(CasosDiariosMD, CasosDiariosCB$num_casos,
                              CasosDiariosAS$num_casos, CasosDiariosGA$num_casos,
                              CasosDiariosCL$num_casos)
colnames(CasosDiariosCCAA) <- c("Fecha", "CasosMD", "CasosCB", "CasosAS",
                              "CasosGA", "CasosCL")
g <- ggplotCovid(CasosDiariosCCAA, aes(Fecha)) + labs(y = "Casos reportados")
g <- g + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                    big.mark=".",
                                                    decimal.mark=",")})
g <- g + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(CasosDiariosCCAA$Fecha,1),
                                by="3 months"),
                    date_minor_breaks = "1 month",
                    date_labels = "%b %Y")
g <- g + scale_color_manual(name="Casos diarios",
                           values = c("Madrid"="blue", "Cantabria"="red",
                                       "Asturias"="deepskyblue",
                                       "Galicia"="seagreen",
                                       "Castilla y León"="magenta"))
g <- g + geom_step(aes(y=CasosMD, color="Madrid"))
g <- g + geom_step(aes(y=CasosCB, color="Cantabria"))
g <- g + geom_step(aes(y=CasosAS, color="Asturias"))
g <- g + geom_step(aes(y=CasosGA, color="Galicia"))
g <- g + geom_step(aes(y=CasosCL, color="Castilla y León"))
g <- g + theme(legend.position=c(.05, .95), legend.justification=c("left", "top"))
g
```



### - Incidencia por cada 100.000 habitantes por CCAA

Como es lógico, los datos no son comparables en términos absolutos por la gran diferencia de población entre estas comunidades autónomas, por no entrar en la forma de vida en unas y otras, aunque sí que haya similitudes entre las de la cornisa cantábrica, y en cómo esto impacta en la dispersión de la enfermedad.

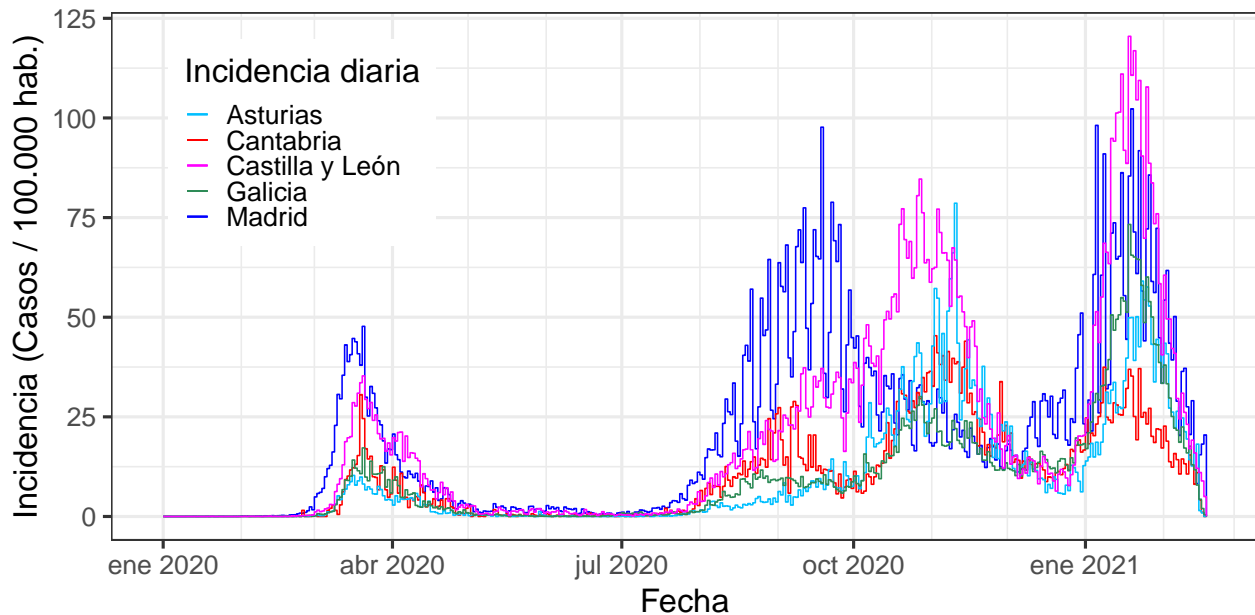
Para solventar este problema representamos ahora número de casos por cada 100.000 habitantes, con los datos de población en cada comunidad disponibles en el momento en el INE (<https://www.ine.es/dynInfo/Infografia/Territoriales/capitulo.html#!tabla>).

```
poblMD <- 6.747E+06
poblCB <- 0.582E+06
poblAS <- 1.019E+06
poblGA <- 2.702E+06
poblCL <- 2.401E+06
IncidenciaCCAA <- data.frame(CasosDiariosCCAA$Fecha,
                             CasosDiariosMD$num_casos/poblMD*1E+05,
                             CasosDiariosCB$num_casos/poblCB*1E+05,
                             CasosDiariosAS$num_casos/poblAS*1E+05,
                             CasosDiariosGA$num_casos/poblGA*1E+05,
                             CasosDiariosCL$num_casos/poblCL*1E+05)
colnames(IncidenciaCCAA) <- c("Fecha", "IncidMD", "IncidCB", "IncidAS",
                             "IncidGA", "IncidCL")
g <- ggplotCovid(IncidenciaCCAA, aes(Fecha))
g <- g + labs(y = "Incidencia (Casos / 100.000 hab.)")
g <- g + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                       big.mark=".",
                                                       decimal.mark=",")})
g <- g + scale_x_date(breaks = seq(as.Date("2020-01-01"),
                                   tail(IncidenciaCCAA$Fecha,1),
                                   by="3 months"),
                     date_minor_breaks = "1 month",
                     date_labels = "%b %Y")
g <- g + scale_color_manual(name="Incidencia diaria",
                           values = c("Madrid"="blue", "Cantabria"="red",
```

```

"Castilla y León"="magenta"))
g <- g + geom_step(aes(y=IncidMD, color="Madrid"))
g <- g + geom_step(aes(y=IncidCB, color="Cantabria"))
g <- g + geom_step(aes(y=IncidAS, color="Asturias"))
g <- g + geom_step(aes(y=IncidGA, color="Galicia"))
g <- g + geom_step(aes(y=IncidCL, color="Castilla y León"))
g <- g + theme(legend.position=c(.05, .95), legend.justification=c("left", "top"))
g

```



#### - Comparación de la IA14 por CCAA y nacional

Con el objetivo de completar la información comparativa entre estas comunidades se adjunta también la incidencia acumulada en 14 días para estas áreas geográficas, junto con la correspondiente al conjunto del territorio nacional:

```

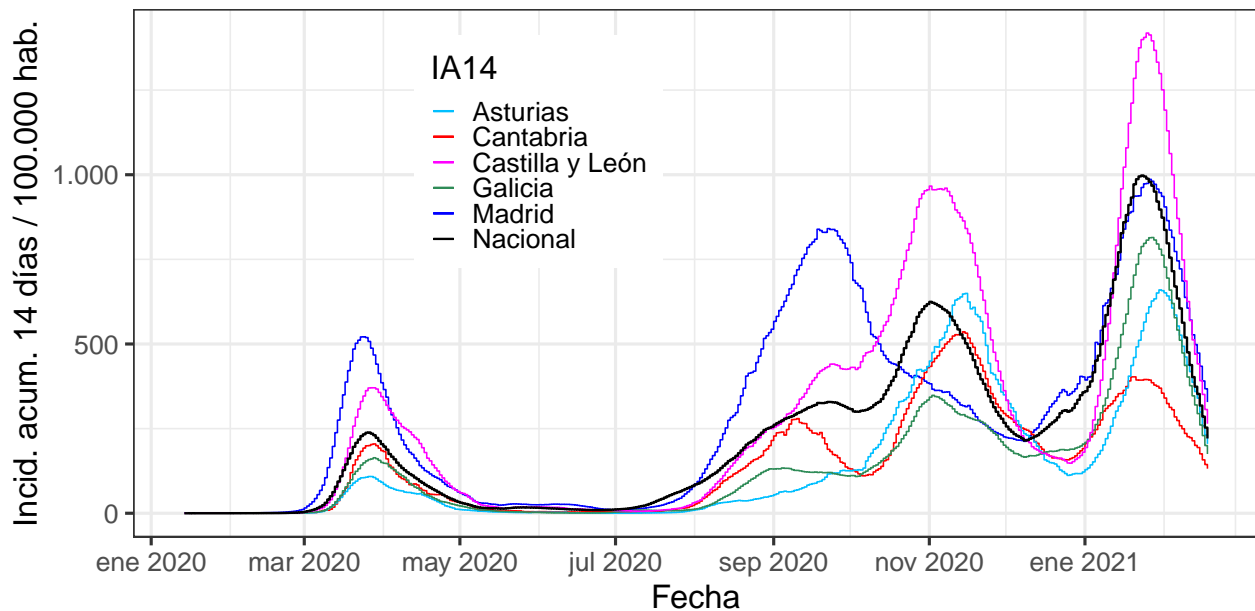
InicioPeriodo14 <- as.Date("2020-01-01")
FinPeriodo14 <- InicioPeriodo14 + 13
IA14_CCAA <- data.frame(Fecha=as.Date(character()), IA14_MD=double(),
                        IA14_CB=double(), IA14_AS=double(), IA14_GA=double(),
                        IA14_CL=double())
while (FinPeriodo14 <= UltFecha) {
  Periodo14 <- seq.Date(
    from = InicioPeriodo14,
    to = FinPeriodo14,
    by = "day")
  DatosIA14_CCAA <- subset(IncidenciaCCAA, Fecha %in% Periodo14)
  IA14_CCAA <- add_row(IA14_CCAA, Fecha = FinPeriodo14,
                      IA14_MD = sum(DatosIA14_CCAA$IncidMD),
                      IA14_CB = sum(DatosIA14_CCAA$IncidCB),
                      IA14_AS = sum(DatosIA14_CCAA$IncidAS),
                      IA14_GA = sum(DatosIA14_CCAA$IncidGA),
                      IA14_CL = sum(DatosIA14_CCAA$IncidCL))
}

```

```

    InicioPeriodo14 <- InicioPeriodo14 + 1
    FinPeriodo14 <- FinPeriodo14 + 1
  }
  ## Añadiendo columna con valores a nivel nacional
  IA14_CCAANac <- add_column(IA14_CCAA, IA14_Nacional=IncidAcum14$IA14)
  ## Gráfico
  g <- ggplotCovid(IA14_CCAANac, aes(Fecha))
  g <- g + labs(y = "Incid. acum. 14 días / 100.000 hab.")
  g <- g + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                         big.mark=".",
                                                         decimal.mark=",")})
  g <- g + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(IA14_CCAANac$Fecha,1),
                                     by="2 months"),
                       date_minor_breaks = "1 month",
                       date_labels = "%b %Y")
  g <- g + scale_color_manual(name = "IA14",
                             values = c("Madrid"="blue", "Cantabria"="red",
                                           "Asturias"="deepskyblue",
                                           "Galicia"="seagreen",
                                           "Castilla y León"="magenta",
                                           "Nacional" = "black"))
  g <- g + geom_step(aes(y=IA14_MD, color="Madrid"))
  g <- g + geom_step(aes(y=IA14_CB, color="Cantabria"))
  g <- g + geom_step(aes(y=IA14_AS, color="Asturias"))
  g <- g + geom_step(aes(y=IA14_GA, color="Galicia"))
  g <- g + geom_step(aes(y=IA14_CL, color="Castilla y León"))
  g <- g + geom_step(aes(y=IA14_Nacional, color="Nacional"), size=0.75)
  g <- g + theme(legend.position=c(.25, .95), legend.justification=c("left", "top"))
  g

```



## 9.- Evolución temporal de la tasa de variación de la incidencia acumulada en 14 días por cada 100.000 habitantes (IA14)

En el siguiente gráfico lo que se representa es cómo varía la incidencia acumulada en 14 días por cada 100.000 habitantes (IA14) expresando la tasa de variación de esta incidencia como:

$$\text{tasaIA14}(i) = (\text{IA14}(i) - \text{IA14}(i-1)) / \text{IA14}(i-1)$$

Se impone la condición para el cálculo de la tasa que IA14(i-1) sea mayor que cero para evitar obtener tasas de crecimiento infinitas por la división con denominador cero y la indeterminación 0/0 en los casos de secuencias de IA14 con valor 0 en los inicios de las series temporales.

Como para el caso de incidencias acumuladas de valores muy bajos, pequeños cambios de la dicha incidencia representan cambios de tasa de evolución muy elevadas al ser el divisor pequeño, modificaremos la condición indicada en el párrafo anterior, exigiendo que la incidencia acumulada en el día anterior tenga como mínimo un valor de 3 casos por cada 100.000 habitantes.

```
InicioIA14 <- as.Date("2020-01-14")
FechaTasa <- as.Date(character())
FechaTasa <- InicioIA14 + 1
tasaEvolIA14 <- data.frame(Fecha=as.Date(character()), tasaIA14_Nac=double(),
                           tasaIA14_MD=double(), tasaIA14_CB=double(),
                           tasaIA14_AS=double(), tasaIA14_GA=double(),
                           tasaIA14_CL=double())
while (FechaTasa <= UltFecha) {
  IA14_CCAANac_Fecha <- subset(IA14_CCAANac, Fecha == FechaTasa)
  IA14_CCAANac_FechaAnt <- subset(IA14_CCAANac, Fecha == FechaTasa-1)
  if(IA14_CCAANac_FechaAnt$IA14_Nacional > 3) {
    tasaIA14auxNac <- signif((IA14_CCAANac_Fecha$IA14_Nacional -
                              IA14_CCAANac_FechaAnt$IA14_Nacional) /
                              IA14_CCAANac_FechaAnt$IA14_Nacional, digits=4)
  } else {tasaIA14auxNac <- 0}
  if(IA14_CCAANac_FechaAnt$IA14_MD > 3) {
    tasaIA14auxMD <- signif((IA14_CCAANac_Fecha$IA14_MD -
                              IA14_CCAANac_FechaAnt$IA14_MD) /
                              IA14_CCAANac_FechaAnt$IA14_MD, digits=4)
  } else {tasaIA14auxMD <- 0}
  if(IA14_CCAANac_FechaAnt$IA14_CB > 3) {
    tasaIA14auxCB <- signif((IA14_CCAANac_Fecha$IA14_CB -
                              IA14_CCAANac_FechaAnt$IA14_CB) /
                              IA14_CCAANac_FechaAnt$IA14_CB, digits=4)
  } else {tasaIA14auxCB <- 0}
  if(IA14_CCAANac_FechaAnt$IA14_AS > 3) {
    tasaIA14auxAS <- signif((IA14_CCAANac_Fecha$IA14_AS -
                              IA14_CCAANac_FechaAnt$IA14_AS) /
                              IA14_CCAANac_FechaAnt$IA14_AS, digits=4)
  } else {tasaIA14auxAS <- 0}
  if(IA14_CCAANac_FechaAnt$IA14_GA > 3) {
    tasaIA14auxGA <- signif((IA14_CCAANac_Fecha$IA14_GA -
                              IA14_CCAANac_FechaAnt$IA14_GA) /
                              IA14_CCAANac_FechaAnt$IA14_GA, digits=4)
  } else {tasaIA14auxGA <- 0}
  if(IA14_CCAANac_FechaAnt$IA14_CL > 3) {
    tasaIA14auxCL <- signif((IA14_CCAANac_Fecha$IA14_CL -
                              IA14_CCAANac_FechaAnt$IA14_CL) /
                              IA14_CCAANac_FechaAnt$IA14_CL, digits=4)
  }
```



```

} else {tasaIA14auxCL <- 0}
tasaEvolIA14 <- add_row(tasaEvolIA14, Fecha = FechaTasa,
                        tasaIA14_Nac = tasaIA14auxNac,
                        tasaIA14_MD = tasaIA14auxMD,
                        tasaIA14_CB = tasaIA14auxCB,
                        tasaIA14_AS = tasaIA14auxAS,
                        tasaIA14_GA = tasaIA14auxGA,
                        tasaIA14_CL = tasaIA14auxCL)

FechaTasa <- FechaTasa + 1
}

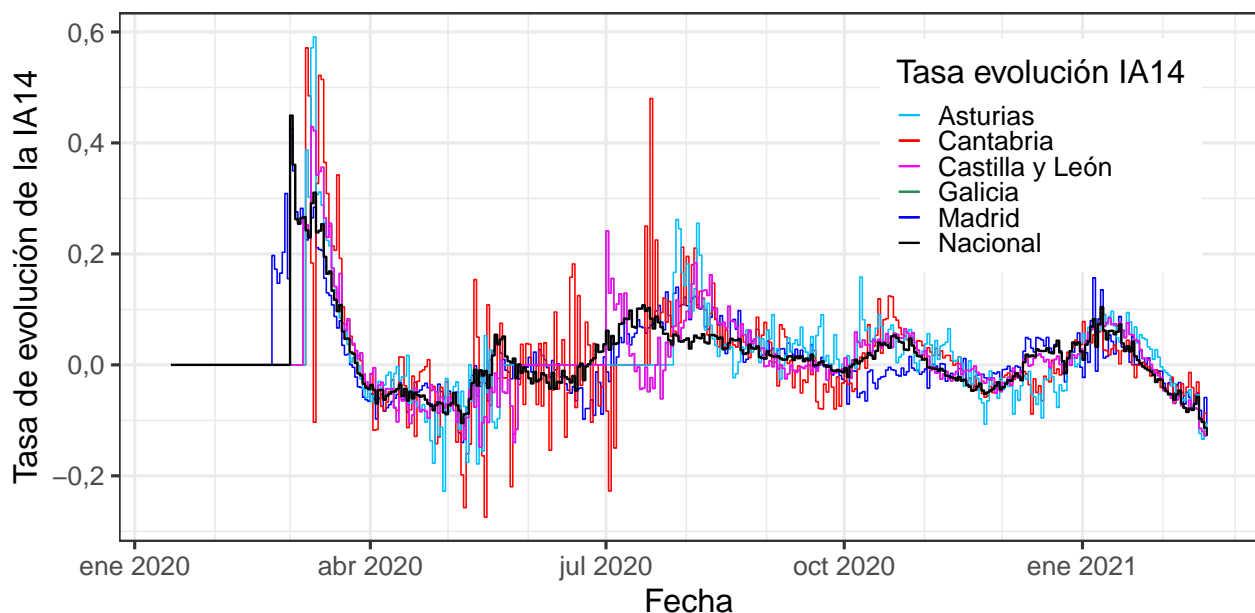
## Gráfico
g <- ggplotCovid(tasaEvolIA14, aes(Fecha))
g <- g + labs(y = "Tasa de evolución de la IA14")
g <- g + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                    big.mark=".",
                                                    decimal.mark=",")})

g <- g + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(tasaEvolIA14$Fecha,1),
                                by="3 months"),
                    date_minor_breaks = "1 month",
                    date_labels = "%b %Y")

g <- g + scale_color_manual(name = "Tasa evolución IA14",
                           values = c("Madrid"="blue", "Cantabria"="red",
                                       "Asturias"="deepskyblue",
                                       "Galicia"="seagreen",
                                       "Castilla y León"="magenta",
                                       "Nacional" = "black"))

g <- g + geom_step(aes(y=tasaIA14_MD, color="Madrid"))
g <- g + geom_step(aes(y=tasaIA14_CB, color="Cantabria"))
g <- g + geom_step(aes(y=tasaIA14_AS, color="Asturias"))
g <- g + geom_step(aes(y=tasaIA14_GA, color="Galicia"))
g <- g + geom_step(aes(y=tasaIA14_GA, color="Castilla y León"))
g <- g + geom_step(aes(y=tasaIA14_Nac, color="Nacional"), size=0.75)
g <- g + theme(legend.position=c(.95, .95), legend.justification=c("right", "top"))
g

```



Hay que hacer notar que, aunque la gráfica resultante tenga una apariencia similar a la del Número reproductivo básico instantáneo - Rt (número promedio de casos secundarios que cada sujeto infectado puede llegar a infectar en una etapa de tiempo (t)), no se trata de este indicador, cuyo cálculo es totalmente diferente al presentado aquí. El número reproductivo básico instantáneo calculado por el Instituto de Salud Carlos III puede ser encontrado en el siguiente enlace: <https://cnecovid.isciii.es/covid19/#ccaa>. Nótese que el nivel de referencia del número reproductivo es 1 mientras que para la tasa de evolución de la IA14 es 0.

Por otro lado es lógica esta similitud entre las gráficas de la tasa de evolución de la IA14 y la del número reproductivo básico instantáneo, ya que números reproductivos altos se corresponden con evoluciones crecientes en la incidencia de la enfermedad mientras que números reproductivos por debajo de 1 marcan evoluciones decrecientes de la incidencia en el número de casos.

## 10.- Datos de mortalidad

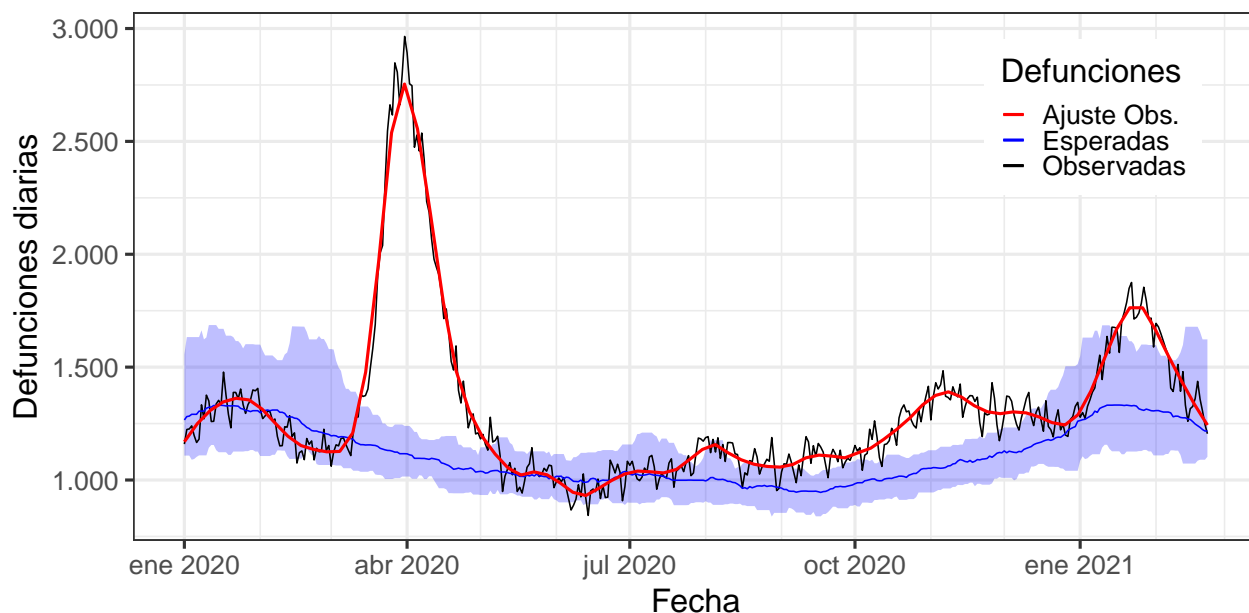
Como siguiente paso del análisis estudiaremos la mortalidad registrada en el periodo de análisis. Los datos se han obtenido del enlace del **Instituto de Salud Carlos III**: <https://momo.isciii.es/public/momo/data>.

```
DatosCompletoMoMo <- read.csv(file.path("data", "data.csv"))
## Descartamos información de detalle de CCAA, sexos y grupos de edad
DatosMoMo <- DatosCompletoMoMo[DatosCompletoMoMo$ambito == "nacional" &
                                DatosCompletoMoMo$cod_sexo == "all" &
                                DatosCompletoMoMo$cod_gedad == "all" &
                                DatosCompletoMoMo$fecha_defuncion > "2019-12-31",
                                c(9:10, 13:15)]
colnames(DatosMoMo) <- c("Fecha", "DefObs", "DefEsp", "DefEsp_q01", "DefEsp_q99")
DatosMoMo <- as.data.table(DatosMoMo)
## Adecuación formato fechas para su correcta lectura
DatosMoMo$Fecha <- as.Date(DatosMoMo$Fecha, format = "%Y-%m-%d")
```

La fecha y hora de descarga de los datos de mortalidad utilizados para la elaboración de los siguientes gráficos y tablas fue (aaaa-mm-dd hh:mm:ss): **2021-02-23 10:01:46**

Representemos en primer lugar la evolución del número de defunciones en comparación con las esperadas y su rango para los percentiles 1 y 99:

```
g <- ggplotCovid(DatosMoMo, aes(Fecha))
g <- g + labs(y = "Defunciones diarias")
g <- g + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                       big.mark=".",
                                                       decimal.mark=",")})
g <- g + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(DatosMoMo$Fecha,1),
                                   by="3 months"),
                     date_minor_breaks = "1 month",
                     date_labels = "%b %Y")
g <- g + scale_color_manual(name = "Defunciones",
                           values = c("Observadas"="black", "Esperadas"="blue",
                                       "Ajuste Obs."="red"))
g <- g + geom_ribbon(aes(ymin=DefEsp_q01, ymax=DefEsp_q99), alpha=0.25, fill="blue")
g <- g + geom_line(aes(y=DefObs, color="Observadas"))
g <- g + geom_line(aes(y=DefEsp, color="Esperadas"))
g <- g + geom_smooth(aes(y=DefObs, color="Ajuste Obs."), formula=y~x, method="loess",
                    span=0.1, method.args=list(degree=2), se=FALSE)
g <- g + theme(legend.position=c(.95, .95), legend.justification=c("right", "top"))
g
```



Se ha agregado una línea de ajuste estadístico de las defunciones observadas para facilitar la visualización de la evolución de las mismas suavizando los dientes de sierra propios de las observaciones por tratarse de datos diarios.

Como se puede ver, existe un periodo de mortalidad totalmente disparada a lo largo de los meses de marzo a mayo, con una punta con valores cercanos al triple de lo esperado, mientras que luego se aprecia otro periodo de desviación al alza, no tan acusado pero más prolongado en el tiempo y con una tendencia creciente a lo largo de varios meses antes de empezar a descender, que cubriría desde agosto hasta bastante avanzado diciembre.

Antes de avanzar en el estudio cuantitativo del exceso de mortalidad haremos un primer análisis cualitativo comparando las curvas de evolución de la incidencia acumulada (IA14) con la curva de defunciones.

Evidentemente lo que primero salta a la vista es la diferencia de proporción del pico de la primera ola con respecto a la segunda ola que ya hemos comentado anteriormente que en lo que respecta a la incidencia acumulada se debe a una subestimación por falta de pruebas diagnósticas, mientras que en lo que concierne a la mortalidad lo que parecen mostrar los datos que ha sucedido es que la primera ola tuvo un altísimo impacto en la población más sensible. No estábamos preparados, faltaba conocimiento y se reaccionó tarde, con lo que las consecuencias fueron desgraciadamente funestas. El confinamiento generalizado, aunque llegase tarde, era necesario y se mostró efectivo. Ahora bien, tanto a la vista de la incidencia acumulada como de la mortalidad no parece justificado de ningún punto que se prolongase por tanto tiempo como se hizo y el desescalamiento debería haber comenzado antes y haber sido con escalones más prolongados en el tiempo para de verdad haber mantenido un control efectivo de la propagación de la enfermedad.

Como era de esperar, tan pronto se levantaron las medidas de restricción, la curva de incidencia empezó a crecer de forma constante. Curiosamente, contrario a lo que se esperaba, el inicio del nuevo curso escolar no tuvo un efecto negativo en la incidencia en el sentido de que ésta se disparase y de hecho en septiembre, primero disminuye el ritmo de crecimiento para luego incluso decrecer la incidencia en la segunda mitad de mes. Por otro lado, y también lógico y esperable, el crecimiento de la incidencia acumulada es acompañado por un crecimiento de la mortalidad que ya en verano supera el percentil 99 de las esperadas. La pregunta inevitable es por qué, con una incidencia creciente una mortalidad superior a la esperada y creciente, no se tomaron medidas antes por parte de las autoridades competentes.

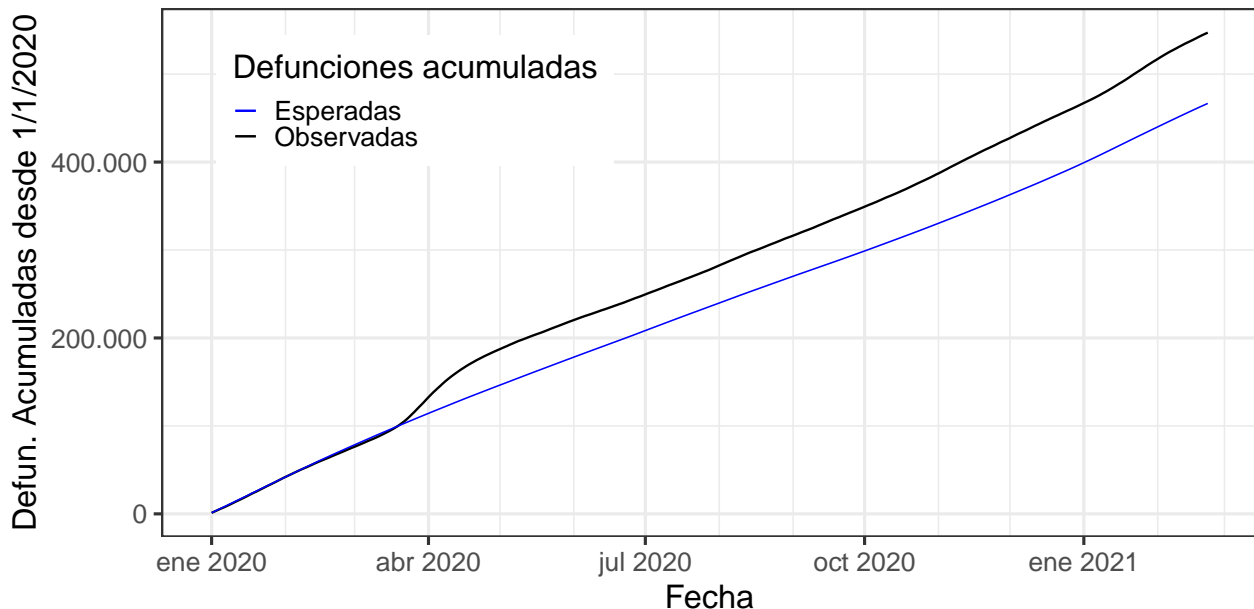
## 11.- Exceso de mortalidad

Técnicamente se define “periodo de exceso de mortalidad” cuando se cumplen las siguientes condiciones:

- Se observa al menos dos días consecutivos con defunciones observadas por encima del percentil 99 de las estimadas.
- La fecha de inicio del periodo es el primer día con las defunciones observadas por encima de las estimadas.
- La fecha de fin del periodo es el último día con las defunciones observadas por encima de las estimadas.
- Si entre la fecha de fin de un periodo y la fecha de inicio del siguiente hay dos días, se unifican ambos periodos, tomando la fecha de inicio del primer periodo y fecha de fin del último.

Con estas premisas podemos aislar los periodos en los que se han producido dichas circunstancias y calcular el exceso de defunciones durante esos lapsos de tiempo concretos. Ahora bien, antes de pasar a realizar dichos cálculos, realicemos uno más básico, comparando directamente las cifras de defunciones esperadas acumuladas a lo largo de 2020 y lo que va de año 2021 con las que realmente se han registrado:

```
DatosMoMo[,AcumObs := cumsum(DefObs)]
DatosMoMo[,AcumEsp := cumsum(DefEsp)]
g <- ggplotCovid(DatosMoMo, aes(Fecha))
g <- g + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                    big.mark=".",
                                                    decimal.mark=",")})
g <- g + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(DatosMoMo$Fecha,1),
                                by="3 months"),
                    date_minor_breaks = "1 month",
                    date_labels = "%b %Y")
g <- g + labs(y = "Defun. Acumuladas desde 1/1/2020")
g <- g + scale_color_manual(name = "Defunciones acumuladas",
                          values = c("Observadas"="black", "Esperadas"="blue"))
g <- g + geom_line(aes(y=AcumObs, color="Observadas"), size = 0.75)
g <- g + geom_line(aes(y=AcumEsp, color="Esperadas"))
g <- g + theme(legend.position=c(.05, .95), legend.justification=c("left", "top"))
g
```



Como podíamos esperar, esta gráfica no aporta gran valor a la hora de la interpretación de la información, más allá del hecho de que las defunciones observadas se distancian de las esperadas de forma muy visible a lo largo de los meses de marzo a mayo de 2020, y que dicho distanciamiento se vuelve a incrementar, ya a menor ritmo, a partir del mes de agosto, aunque se aprecia que vuelve a repuntar ligeramente en noviembre de 2020, con la punta de la segunda ola, y otro repunte más en enero con la tercera ola de la pandemia.

```
## Acumulados totales desde inicio de 2020
TotalDefAcumObs <- last(DatosMoMo$AcumObs)
TotalDefAcumEsp <- round(last(DatosMoMo$AcumEsp))
TotalExcesoDef <- TotalDefAcumObs - TotalDefAcumEsp
PorcentajeExcesoDef <- round(TotalExcesoDef/TotalDefAcumEsp*100, 1)
## Acumulados desde inicio de 2020 hasta 21 de junio de 2020
DefAcumObs21jun <- DatosMoMo$AcumObs[DatosMoMo$Fecha == "2020-06-21"]
DefAcumEsp21jun <- round(DatosMoMo$AcumEsp[DatosMoMo$Fecha == "2020-06-21"])
ExcesoDef21jun <- DefAcumObs21jun - DefAcumEsp21jun
PorcentExcesoDef21jun <- round(ExcesoDef21jun/DefAcumEsp21jun*100, 1)
```

Más interesante resulta la comparación directa de las cifras acumuladas hasta la fecha. En este caso tenemos, con los datos disponibles, un total de **547.127** defunciones observadas y **466.676** defunciones esperadas, resultando un exceso de **80.451** defunciones. Expresando dicho exceso en términos porcentuales, nos encontramos con un **17,2 %** más fallecimientos de los esperados.

Por afán de completar la visión de la evolución de estas variables, presentamos a continuación esos mismos valores en la fecha en la que se levantó el estado de alarma, 21 de junio de 2020, buscando una fecha en lo que podríamos denominar “**final de la primera ola**”, que no “*la derrota de la pandemia*” (sic):

- Defunciones acumuladas observadas: **239.309** personas
- Defunciones acumuladas esperadas: **198.236** personas
- Exceso de defunciones: **41.073** personas
- En tanto por ciento: **20,7 %**

Retomando la senda de la ortodoxia y aplicando ahora sí los criterios técnicos “oficiales” que presentábamos más arriba que definen los periodos de exceso de mortalidad, las fechas que delimitan el principio y final de los periodos de exceso padecidos a lo largo de 2020 son (fechas en formato aaaa-mm-dd):

```
## Columnas auxiliares variables buleanas
DatosMoMo[, MQEsp := DefObs>DefEsp]
DatosMoMo[, MQEsp99_2dias := {DefObs>DefEsp_q99 & lag(DefObs)>lag(DefEsp_q99)}]
## Determinar inicio y fin de periodos de exceso
PeriodosExceso <- data.table("Inicio"=as.Date(character()), "Fin"=as.Date(character()))
i <- as.integer(1)
while (i < nrow(DatosMoMo)) { ## Recorrer tabla
  if (DatosMoMo$MQEsp99_2dias[i] == TRUE) { ## Encontrar dos días > quant99(Esp)
    j <- as.integer(1)
    while (DatosMoMo$MQEsp[i-j] == TRUE && DatosMoMo$MQEsp[i-j-1] == TRUE) {
      j <- j + 1
    }
    PeriodosExcesoInicioN <- as.Date(DatosMoMo$Fecha[i-j])
    for (k in 0:(nrow(DatosMoMo)-i)) {
      if (i + k == nrow(DatosMoMo)) {
        PeriodosExcesoFinN <- as.Date(DatosMoMo$Fecha[i+k])
        PeriodosExceso <- add_row(PeriodosExceso,
                                   Inicio = PeriodosExcesoInicioN,
                                   Fin = PeriodosExcesoFinN)
      }
      i <- i + k
      break
    }
    if (DatosMoMo$MQEsp[i+k] == TRUE &&
        DatosMoMo$MQEsp[i+k+1] == FALSE) {
      PeriodosExcesoFinN <- as.Date(DatosMoMo$Fecha[i+k])
      PeriodosExceso <- add_row(PeriodosExceso,
                                 Inicio = PeriodosExcesoInicioN,
                                 Fin = PeriodosExcesoFinN)
    }
  }
  i <- i + 1
}
```

```

        i <- i + k
        break
      }
    }
  }
  i <- i + 1
}
## Unir periodos de exceso próximos (adyacentes o un día intermedio)
PeriodosExcesoUnido <- data.table("Inicio"=as.Date(character()),
                                   "Fin"=as.Date(character()))
n <- as.integer(1)
while (n <= nrow(PeriodosExceso)) {
  PeriodosExcesoUnidoInicioN <- PeriodosExceso$Inicio[n]
  while (((PeriodosExceso$Inicio[n+1] - PeriodosExceso$Fin[n]) <= 2) &
        n < nrow(PeriodosExceso)) {n <- n + 1}
  PeriodosExcesoUnidoFinN <- PeriodosExceso$Fin[n]
  PeriodosExcesoUnido <- add_row(PeriodosExcesoUnido,
                                Inicio = PeriodosExcesoUnidoInicioN,
                                Fin = PeriodosExcesoUnidoFinN)
  n <- n + 1
}

```

- Antes de unificar periodos de exceso próximos:

Inicio	Fin
2020-03-10	2020-05-09
2020-07-20	2020-08-29
2020-09-01	2020-12-25
2020-12-28	2021-02-13

- Después de unificar los periodos de exceso cercanos (adyacentes o con un día intermedio):

Inicio	Fin
2020-03-10	2020-05-09
2020-07-20	2020-08-29
2020-09-01	2020-12-25
2020-12-28	2021-02-13

```

## Excesos de defunciones durante los periodos de exceso
PeriodoExcesoN <- as.Date(character())
ExcesoDefunciones <- data.table("Exceso"=integer())
DefuncionesEsperadas <- data.table("Esperadas"=integer())
for (i in 1:nrow(PeriodosExcesoUnido)) {
  PeriodoExcesoN <- seq.Date(
    from = as.Date(PeriodosExcesoUnido$Inicio[i]),
    to = as.Date(PeriodosExcesoUnido$Fin[i]),
    by = "day")
  DatosMoMoPeriodoN <- subset(DatosMoMo, Fecha %in% PeriodoExcesoN)
  ExcesoDefunciones <- add_row(ExcesoDefunciones, "Exceso" =
    (sum(DatosMoMoPeriodoN$DefObs)
     -round(sum(DatosMoMoPeriodoN$DefEsp))))
}

```

```
DefuncionesEsperadas <- add_row(DefuncionesEsperadas, "Esperadas" =
                                round(sum(DatosMoMoPeriodoN$DefEsp)))
}
PeriodosExcesoUnido <- add_column(PeriodosExcesoUnido, "Exceso de defunciones" =
                                ExcesoDefunciones$Exceso)
TotalExcesoDefunciones <- sum(PeriodosExcesoUnido$"Exceso de defunciones")
```

Los excesos de defunciones en estos 4 periodos son:

Inicio	Fin	Exceso de defunciones
2020-03-10	2020-05-09	44.550
2020-07-20	2020-08-29	4.673
2020-09-01	2020-12-25	21.489
2020-12-28	2021-02-13	12.256

Siendo el total agregado de exceso de defunciones de **82.968** personas.

Expresándolo en términos porcentuales, el exceso de defunciones es un **17,8 %** superior al total de las esperadas hasta la fecha. Como es lógico, este valor porcentual se irá reduciendo a medida que transcurra el tiempo desde el final del último episodio de exceso de defunciones.

Para eliminar esta dependencia temporal, veamos estos excesos en términos porcentuales con respecto a las esperadas, pero circunscritos exclusivamente al propio periodo de exceso de defunciones y dejando fuera el resto de la serie temporal:

```
PeriodosExcesoUnido <- add_column(PeriodosExcesoUnido, "Porcentaje de exceso" =
                                round((PeriodosExcesoUnido$"Exceso de defunciones" /
                                DefuncionesEsperadas$Esperadas * 100), 1))
```

Inicio	Fin	Exceso de defunciones	Porcentaje de exceso
2020-03-10	2020-05-09	44.550	66,8
2020-07-20	2020-08-29	4.673	11,5
2020-09-01	2020-12-25	21.489	17,7
2020-12-28	2021-02-13	12.256	19,6

Aunque en el exceso de defunciones haya casos de fallecimiento no directamente imputables a la COVID-19, hay que asignar dichas muertes a la crisis del COVID-19. Si determinadas patologías no son debidamente atendidas en tiempo y forma por la sobrecarga del sistema sanitario provocada por la pandemia, los fallecimientos asociados a las mismas son por tanto atribuibles a la COVID-19 aunque el virus no haya sido la causa directa del fallecimiento correspondiente.

El índice de mortalidad de la COVID-19 en España en el periodo de estudio, medido como exceso de mortalidad atribuible directa o indirectamente a la COVID por cada mil habitantes, es de **1,75**.

Antes de seguir avanzando no podemos dejar de llamar la atención sobre el hecho de que en la determinación de las cifras de exceso de defunciones se ha utilizado como nivel de referencia el número de defunciones esperadas. Es perfectamente argumentable que durante el periodo de estado de alarma este nivel de comparación debería ser inferior al estadísticamente obtenido con datos de años previos ya que el propio estado de alarma tuvo por necesidad incidencia en el número de fallecimientos por accidente laboral y por accidente de tráfico, sin duda disminuyéndolos. Consecuentemente debería rebajarse el patrón de referencia de defunciones esperadas durante el estado de confinamiento y el exceso de defunciones por causa de la COVID-19 sería superior al mostrado más arriba. Aunque es posible realizar estimaciones de estas desviaciones con datos disponibles públicamente, dejamos esa posibilidad de perfeccionamiento del estudio para mejor oportunidad.

## 12.- Mortalidad acumulada en 14 días y comparación con la Incidencia acumulada en 14 días

Al hablar de la incidencia diaria de la pandemia veíamos que su valor se veía claramente influenciado por el menor número de pruebas realizadas durante los fines de semana y festivos, generando unos evidentes dientes de sierra a la hora de su representación gráfica. Aunque la razón de la variabilidad en las defunciones observadas no es la misma, recurriremos a una transformación similar, sumando defunciones en periodos móviles de 14 días y dividiendo la cantidad resultante por la población en España, obteniendo una variable que denominaremos mortalidad acumulada en 14 días por cada millón de habitantes (MA14).

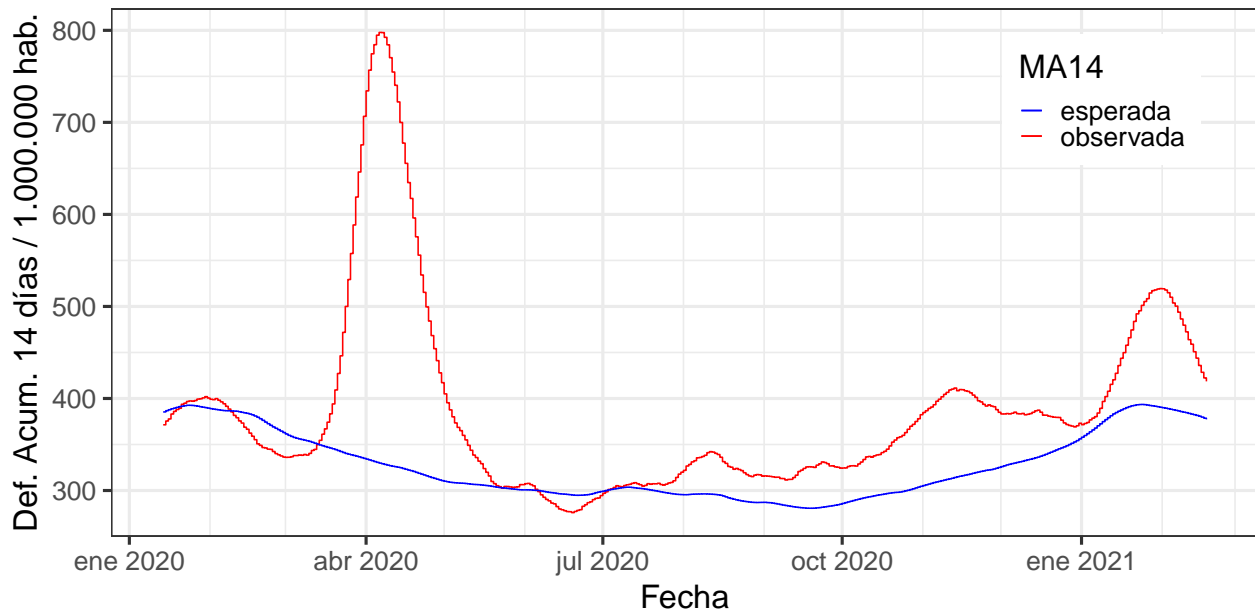
Representando gráficamente la MA14 de las defunciones observadas y la MA14 de las defunciones esperadas, observamos que mantienen la semejanza con los datos de origen, con la lógica distorsión de escala y desplazamiento temporal, y que se ha conseguido el deseado efecto de suavizar los dientes de sierra en lo que respecta a las defunciones observadas.

```
## Inicialización de variables
InicioPeriodo14 <- as.Date("2020-01-01")
FinPeriodo14 <- InicioPeriodo14 + 13
MortAcum14 <- data.frame(Fecha=as.Date(character()), MA14obs=double(),
                          MA14esp=double())

## Mortalidades acumuladas en 14 días - observada y esperada
while (FinPeriodo14 <= UltFecha) {
  Periodo14 <- seq.Date(
    from = InicioPeriodo14,
    to = FinPeriodo14,
    by = "day")
  DatosMA14 <- subset(DatosMoMo, Fecha %in% Periodo14)
  MortAcum14 <- add_row(MortAcum14, Fecha = FinPeriodo14,
                        MA14obs = sum(DatosMA14$DefObs)/poblESP*1E06,
                        MA14esp = sum(DatosMA14$DefEsp)/poblESP*1E06)
  InicioPeriodo14 <- InicioPeriodo14 + 1
  FinPeriodo14 <- FinPeriodo14 + 1
}

## Gráfico - Mortalidades acumuladas 14 días - observada y esperada
g <- ggplotCovid(MortAcum14, aes(Fecha))
g <- g + labs(y = "Def. Acum. 14 días / 1.000.000 hab.")
g <- g + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                       big.mark=".",
                                                       decimal.mark=",")})
g <- g + scale_x_date(breaks = seq(as.Date("2020-01-01"), tail(MortAcum14$Fecha,1),
                                   by="3 months"),
                     date_minor_breaks = "1 month",
                     date_labels = "%b %Y")
g <- g + scale_color_manual(name = "MA14",
                            values = c("observada" = "red", "esperada" = "blue"))
g <- g + geom_step(aes(y=MA14obs, color = "observada"))
g <- g + geom_step(aes(y=MA14esp, color = "esperada"))
g <- g + theme(legend.position=c(.95, .95), legend.justification=c("right", "top"))
g
```





Ahora bien, lo que realmente nos interesa es comparar la diferencia entre estas variables de mortalidades acumuladas, observadas menos esperadas, que denominaremos exceso de mortalidad acumulada (EMA14), con la de la incidencia acumulada (IA14). Veamos qué pasa cuando escalamos ambas, EMA14 e IA14, y las representamos en el mismo gráfico:

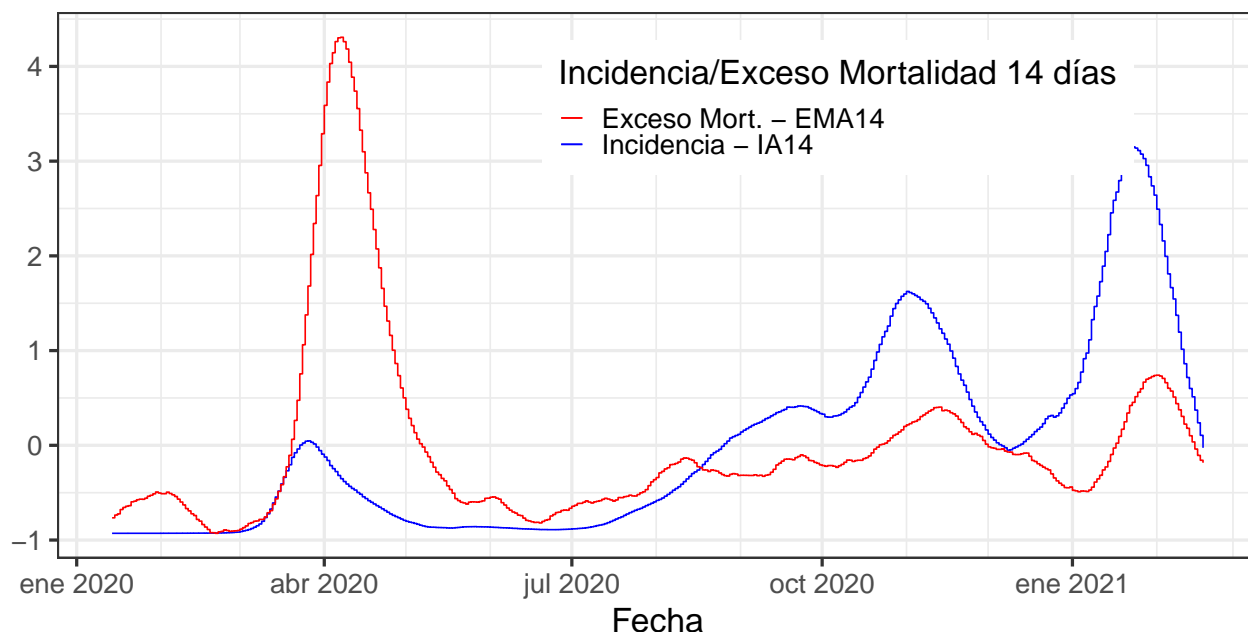
```
## Preparando tabla de datos
IA14vsEMA14 <- cbind(IncidAcum14, EMA14 = MortAcum14$MA14obs - MortAcum14$MA14esp)

## Gráfico - IA14 vs EMA14
g <- ggplotCovid(IA14vsEMA14, aes(Fecha))
g <- g + labs(y = "")
g <- g + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                    big.mark=".",
                                                    decimal.mark=",")})

g <- g + scale_x_date(breaks = seq(as.Date("2020-01-01"),
                                tail(IA14vsEMA14$Fecha,1),
                                by="3 months"),
                    date_minor_breaks = "1 month",
                    date_labels = "%b %Y")

g <- g + scale_color_manual(name = "Incidencia/Exceso Mortalidad 14 días",
                          values = c("Incidencia - IA14" = "blue",
                                      "Exceso Mort. - EMA14" = "red"))

g <- g + geom_step(aes(y = scale(IA14), color = "Incidencia - IA14"))
g <- g + geom_step(aes(y = scale(EMA14), color = "Exceso Mort. - EMA14"))
g <- g + theme(legend.position=c(.65, .95), legend.justification=c("center", "top"))
g
```



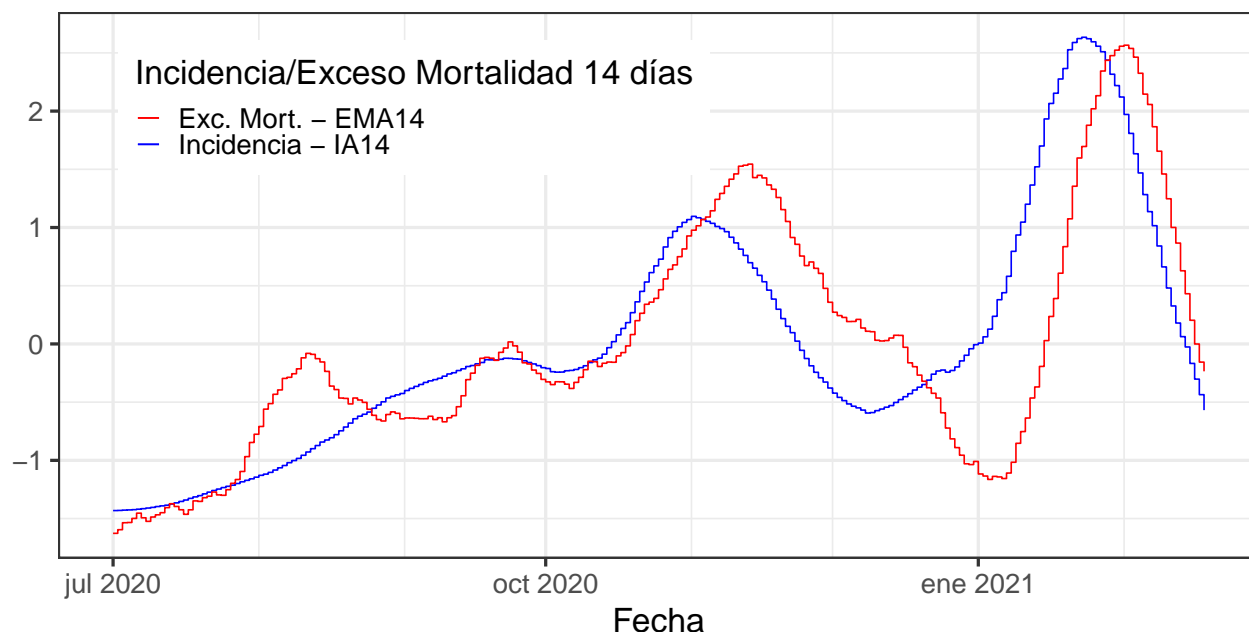
Al comparar ambas gráficas vuelve a resaltar el que se invierta la proporción entre el pico de la primera ola y el de la segunda, resultado de la escasez de ensayos en la primera mitad de 2020, sin poder menoscar las mejoras en los tratamientos de la enfermedad. En vez de intentar estimar la IA14 durante es primera mitad de año con conjeturas falibles, lo que haremos como siguiente paso es simplemente descartar el periodo desde el 1 de enero de 2020 hasta el 30 de junio de 2020 y analizar los datos correspondientes a fechas posteriores.

```
## Preparando tabla de datos
IA14vsEMA14s2 <- subset(IA14vsEMA14, Fecha > as.Date("2020-06-30"))
## Gráfico - IA14 vs EMA14 desde 01/07/2020
g <- ggplotCovid(IA14vsEMA14s2, aes(Fecha))
g <- g + labs(y = "")
g <- g + scale_y_continuous(labels=function(n){format(n, scientific=FALSE,
                                                    big.mark=".",
                                                    decimal.mark=",")})

g <- g + scale_x_date(breaks = seq(as.Date("2020-07-01"),
                                tail(IA14vsEMA14s2$Fecha,1),
                                by="3 months"),
                    date_minor_breaks = "1 month",
                    date_labels = "%b %Y")

g <- g + scale_color_manual(name = "Incidencia/Exceso Mortalidad 14 días",
                          values = c("Incidencia - IA14" = "blue",
                                      "Exc. Mort. - EMA14" = "red"))

g <- g + geom_step(aes(y = scale(IA14), color = "Incidencia - IA14"))
g <- g + geom_step(aes(y = scale(EMA14), color = "Exc. Mort. - EMA14"))
g <- g + theme(legend.position=c(.05, .95), legend.justification=c("left", "top"))
g
```



La correlación que ya se adivinaba con datos desde el 1 de enero de 2020 se hace mucho más evidente con la serie arrancando el 1 de julio de 2020, además de verificarse el esperado decalaje en el tiempo entre los picos de ambas variables, alcanzándose el máximo de exceso de mortalidad después de que la incidencia acumulada empiece ya a decrecer y el inicio del crecimiento en el exceso de mortalidad se produce con demora respecto al inicio del ascenso en la incidencia, por el tiempo de desarrollo de la enfermedad hasta llegar a consecuencias fatales.

Nota: No perdemos de vista en este apartado y en el siguiente que estamos comparando una variable que sí que está directamente correlacionada con el virus responsable de la COVID-19, porque son identificaciones del virus mediante pruebas médicas, con defunciones de todo tipo y causa. Esto plantea una debilidad de base en los resultados obtenidos y en conclusiones que se infieran de éstos, pero la magnitud de la incidencia de la pandemia es tal que por los propios resultados obtenidos esta deficiencia de origen se puede obviar para cierto tipo de conclusiones de carácter más general. Evidentemente esta deficiencia es inasumible para un estudio epidemiológico serio pero no para el tipo de análisis que se pretende con este documento, que no deja de ser informal pese a una mantener una estructura ordenada y con ciertas semejanzas con un artículo científico.

### 13.- Análisis por grupos de edad y sexo de las defunciones ocurridas durante la primera ola de la pandemia

En este apartado haremos una somera disección de los fallecimientos producidos durante la primera ola de la pandemia por grupos de edad y por sexo. Tomaremos como periodo de referencia para el análisis de datos el correspondiente al primer periodo de exceso de defunciones tal y como se definió y determinó más arriba.

En resumen, los datos que se presentan ahora corresponden al periodo entre el **10 de marzo de 2020** y el **09 de mayo de 2020**.

En primera instancia nos limitaremos a analizar los datos para el agregado nacional, sin entrar en el detalle de lo acontecido en cada comunidad autónoma.

```
Periodo01a1 <- as.Date(character())
Periodo01a1 <- seq.Date(from = as.Date(PeriodosExcesoUnido$Inicio[1]),
                        to = as.Date(PeriodosExcesoUnido$Fin[1]),
                        by = "day")
DatosMoMo01a1 <- subset(DatosCompletoMoMo, as.Date(fecha_defuncion) %in%
                        Periodo01a1 & DatosCompletoMoMo$ambito == "nacional",
                        select = c(6, 8:10))
```

```

colnames(DatosMoMo01a1) <- c("sexo", "edad", "fecha", "defunciones")
DatosMoMo01a1 <- as.data.table(DatosMoMo01a1)
DatosMoMo01a1$fecha <- as.Date(DatosMoMo01a1$fecha, format = "%Y-%m-%d")
DatosMoMo01a1$sexo <- factor(DatosMoMo01a1$sexo, levels = c("hombres", "mujeres",
"todos"))
DatosMoMo01a1$edad <- factor(DatosMoMo01a1$edad, levels = c("edad < 65",
"edad 65-74",
"edad > 75",
"todos"))
Tabla01a1 <- xtabs(defunciones ~ sexo + edad, data = DatosMoMo01a1)

```

	edad < 65	edad 65-74	edad > 75	todos
hombres	7.645	9.681	38.489	55.815
mujeres	3.896	4.575	45.906	54.377
todos	11.773	14.438	85.042	111.253

Nótese que los desgloses por sexo siempre suman una cantidad inferior al total agregado correspondiente. Esto no es un error de programación sino una deficiencia, característica intrínseca por decirlo de una forma eufemística, de los datos de partida, en los que para una fecha dada las sumas por sexos no llegan al “todos” correspondiente, ni en el agregado total ni en los subtotales por franjas de edad.

Ante la disyuntiva de cómo continuar después de detectada esta discordancia, en vez de quedarnos con los números que representan las agregaciones “todos” (fila inferior y columna derecha), daremos por buenos los datos de detalle y calcularemos los nuevos agregados parciales y total.

```

DatosMoMo01a1 <- DatosMoMo01a1[sexo != "todos",]
DatosMoMo01a1 <- DatosMoMo01a1[edad != "todos",]
DatosMoMo01a1$sexo <- factor(DatosMoMo01a1$sexo, levels = c("hombres", "mujeres"))
DatosMoMo01a1$edad <- factor(DatosMoMo01a1$edad, levels = c("edad < 65",
"edad 65-74",
"edad > 75"))
Tabla01a1 <- addmargins(xtabs(defunciones ~ sexo + edad, data = DatosMoMo01a1))
Tabla01a1Porc <- addmargins(prop.table(xtabs(defunciones ~ sexo + edad, data =
DatosMoMo01a1))*100)

```

La nueva tabla de distribución por sexos y edad queda entonces de la siguiente manera:

Expresándolo en número de defunciones:

	edad < 65	edad 65-74	edad > 75	Sum
hombres	7.645	9.681	38.489	55.815
mujeres	3.896	4.575	45.906	54.377
Sum	11.541	14.256	84.395	110.192

Representándolo como porcentajes:

	edad < 65	edad 65-74	edad > 75	Sum
hombres	6,94	8,79	34,9	50,7
mujeres	3,54	4,15	41,7	49,3
Sum	10,47	12,94	76,6	100,0

Sobre esta tabla hay que remarcar el hecho de que estamos obteniendo el total de fallecimientos, con independencia de la causa, no sólo por la COVID-19, con lo cual por sí sola no puede arrojar mucha luz ya que no podemos separar los casos “Covid” de los “no-Covid”. Lo que haremos para solventar esta carencia es comparar el reparto porcentual durante la ola con un periodo de referencia. En este caso no nos complicaremos con medias de largos periodos y lo compararemos con la distribución correspondiente al año 2019.

```
Periodo2019 <- as.Date(character())
Periodo2019 <- seq.Date(from = as.Date("2019-01-01"),
                        to = as.Date("2019-12-31"),
                        by = "day")
DatosMoMo2019 <- subset(DatosCompletosMoMo, as.Date(fecha_defuncion) %in%
                        Periodo2019 & DatosCompletosMoMo$ambito == "nacional",
                        select = c(6, 8:10))
colnames(DatosMoMo2019) <- c("sexo", "edad", "fecha", "defunciones")
DatosMoMo2019 <- as.data.table(DatosMoMo2019)
DatosMoMo2019$fecha <- as.Date(DatosMoMo2019$fecha, format = "%Y-%m-%d")
DatosMoMo2019$sexo <- factor(DatosMoMo2019$sexo, levels = c("hombres", "mujeres",
                                                           "todos"))
DatosMoMo2019$edad <- factor(DatosMoMo2019$edad, levels = c("edad < 65",
                                                           "edad 65-74",
                                                           "edad > 75",
                                                           "todos"))
Tabla2019 <- xtabs(defunciones ~ sexo + edad, data = DatosMoMo2019)
```

	edad < 65	edad 65-74	edad > 75	todos
hombres	31.401	31.798	111.282	174.481
mujeres	15.975	15.394	133.797	165.166
todos	49.535	48.432	249.080	347.047

Como ya apuntábamos antes, el problema de falta de coherencia entre los grupos “todos” y sus desgloses es inherente a los propios datos y volveremos a depurarlos de la misma manera en esta ocasión, conservando los desgloses y calculando nuevos subtotaes y total agregado:

```
DatosMoMo2019 <- DatosMoMo2019[sexo != "todos",]
DatosMoMo2019 <- DatosMoMo2019[edad != "todos",]
DatosMoMo2019$sexo <- factor(DatosMoMo2019$sexo, levels = c("hombres", "mujeres"))
DatosMoMo2019$edad <- factor(DatosMoMo2019$edad, levels = c("edad < 65",
                                                           "edad 65-74",
                                                           "edad > 75"))
Tabla2019 <- addmargins(xtabs(defunciones ~ sexo + edad, data = DatosMoMo2019))
Tabla2019Porc <- addmargins(prop.table(xtabs(defunciones ~ sexo + edad, data =
                                             DatosMoMo2019))*100)
```

En número de fallecimientos:

	edad < 65	edad 65-74	edad > 75	Sum
hombres	31.401	31.798	111.282	174.481
mujeres	15.975	15.394	133.797	165.166
Sum	47.376	47.192	245.079	339.647

Como porcentajes:

	edad < 65	edad 65-74	edad > 75	Sum
hombres	9,25	9,36	32,8	51,4
mujeres	4,70	4,53	39,4	48,6
Sum	13,95	13,89	72,2	100,0

Coloquemos las tablas de porcentajes juntas una con la otra para que sea fácil compararlas:

- Primera ola de la pandemia:

	edad < 65	edad 65-74	edad > 75	Sum
hombres	6,94	8,79	34,9	50,7
mujeres	3,54	4,15	41,7	49,3
Sum	10,47	12,94	76,6	100,0

- Año 2019:

	edad < 65	edad 65-74	edad > 75	Sum
hombres	9,25	9,36	32,8	51,4
mujeres	4,70	4,53	39,4	48,6
Sum	13,95	13,89	72,2	100,0

Como se puede ver, con estos datos no es posible afirmar que la COVID-19 en términos de mortalidad haya afectado significativamente más a la población masculina que a la femenina y sólo se puede apreciar un desplazamiento de los fallecimientos hacia las franjas de edad mayores, algo que por otro lado es esperable puesto que la enfermedad afecta más severamente a las personas con patologías previas y éstas se encuentran por lógica mayoritariamente entre los grupos de edad más avanzada.

.....

## Apéndice - Referencias

- (1) RStudio Team (2021). RStudio: Integrated Development Environment for R. RStudio, PBC, Boston, MA. URL: <http://www.rstudio.com/>
- (2) R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL: <https://www.R-project.org/>
- (3) Garrett Golemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL: <http://www.jstatsoft.org/v40/i03/>
- (4) Yihui Xie (2020). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.30.
- (5) Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, <https://doi.org/10.21105/joss.01686>
- (6) H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.
- (7) Matt Dowle and Arun Srinivasan (2020). data.table: Extension of `data.frame`. R package version 1.13.2. <https://CRAN.R-project.org/package=data.table>