

# Análisis de la evolución de la incidencia de la COVID-19 en España

Juan Matorras Díaz-Caneja

30/11/2020

## Introducción

Este es un ejercicio básico de análisis de los datos de la incidencia de la COVID-19 en España a lo largo de 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 he iniciado en la segunda mitad de septiembre de 2020.

Los datos de partida son los publicados por el Gobierno de España en la web **datos.gob.es** a través del siguiente enlace: <https://datos.gob.es/es/catalogo/e05070101-evolucion-de-enfermedad-por-el-coronavirus-covid-19>.

El grueso del informe se centra sobre los totales en España agregando los datos disponibles por Comunidades Autónomas, aunque también se muestran información de las CCAA de Madrid y Cantabria. La razón de la selección de estas dos comunidades y no otras es simple y llanamente que nació y crecí en la última, manteniendo allí vínculos familiares y de amistad, mientras que en la primera he pasado prácticamente la mitad de mi vida, sigo viviendo en ella y previsiblemente así seguirá siendo en los próximos años.

## Proceso metodológico y software utilizado

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.

```
DatosCompletos <- read.csv(file.path("data", "datos_ccaas.csv"))
## Descartamos el método de detección (PCR, antígenos,...)
DatosCCAAs <- DatosCompletos[, 1:3]
## Adecuación formato fechas para su correcta lectura
DatosCCAAs$fecha <- as.Date(DatosCCAAs$fecha, format = "%Y-%m-%d")
## Agrupación de información de CCAA para datos a nivel nacional
Datos <- DatosCCAAs %>% 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 las tablas y gráficos incluidos en este informe ha sido (aaaa-mm-dd hh:mm:ss): **2020-11-30 15:15:05**

El análisis se ha llevado a cabo utilizando el software libre para análisis estadístico **R**, versión 4.0.2. (1)

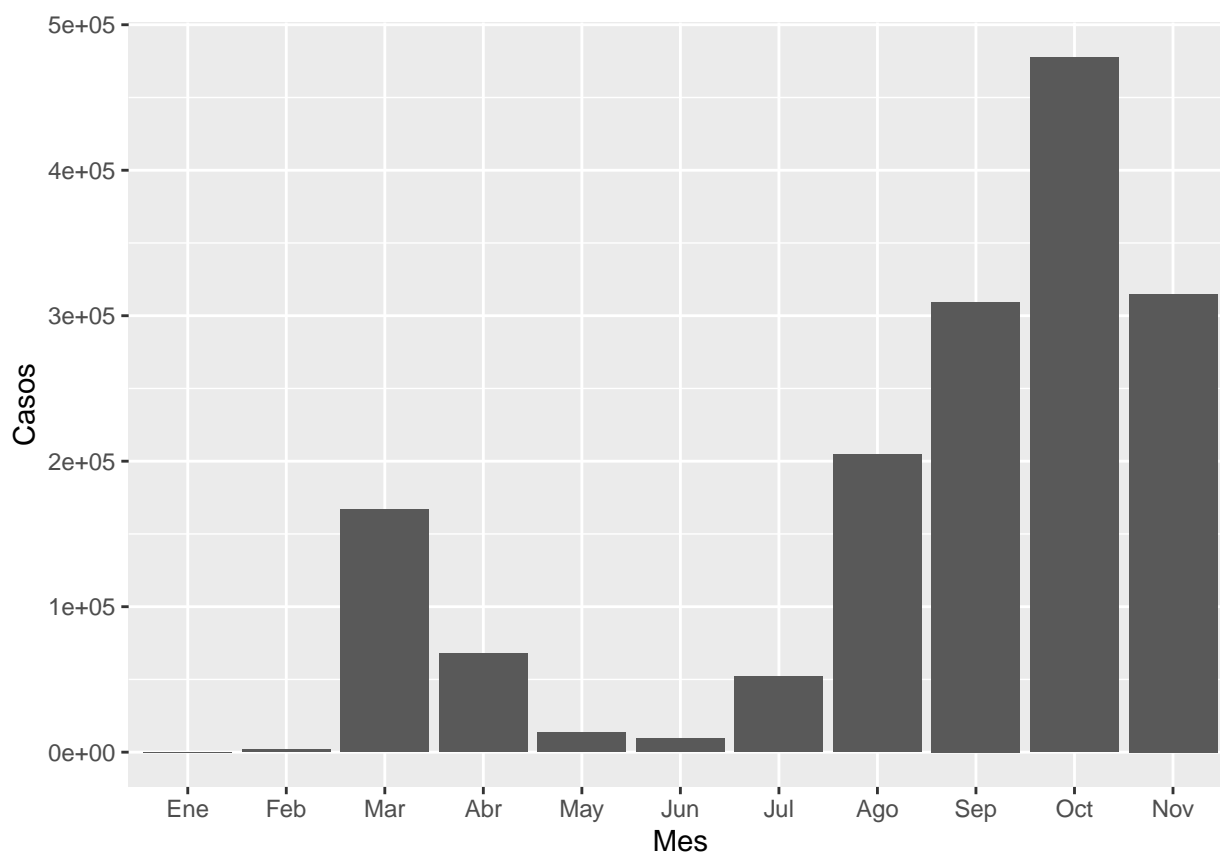
Se ha hecho uso también de los paquetes complementarios:

- **lubridate** para facilitar el manejo de fechas. (2)
- **knitr** para mejorar la apariencia de tablas. (3)
- **tidyverse** por los paquetes que incluye para ayudar en la extracción de la información y los gráficos mejorados de **ggplot2**. (4)
- **data.table** con el objeto de manipular las tablas más eficientemente. (5)

## Incidencia mensual 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:

```
CasosMensuales <- tapply(Datos$num_casos, month(Datos$fecha), sum)
TablaCasosMensuales <- data.frame(head(Meses, length(CasosMensuales)),
  format(CasosMensuales, big.mark = ".", decimal.mark = ","))
colnames(TablaCasosMensuales) <- c("Mes", "Nº casos mensuales")
TotalCasosOrigen <- sum(Datos$num_casos)
PorcentajePoblacion <- paste(format(TotalCasosOrigen / poblESP *100, digits = 4,
  decimal.mark = ","), "%")
dfCasosMensuales <- data.frame(Mes = head(Meses, length(CasosMensuales)),
  Casos = CasosMensuales)
dfCasosMensuales$Mes <- factor(dfCasosMensuales$Mes, levels = dfCasosMensuales$Mes)
d <- ggplot(dfCasosMensuales, aes(Mes, Casos))
d + geom_col()
```



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

Mes	Nº casos mensuales
Ene	45
Feb	1.745
Mar	166.759
Abr	67.800
May	13.306
Jun	9.423
Jul	52.186
Ago	204.433

Mes	Nº casos mensuales
Sep	309.355
Oct	477.610
Nov	314.842

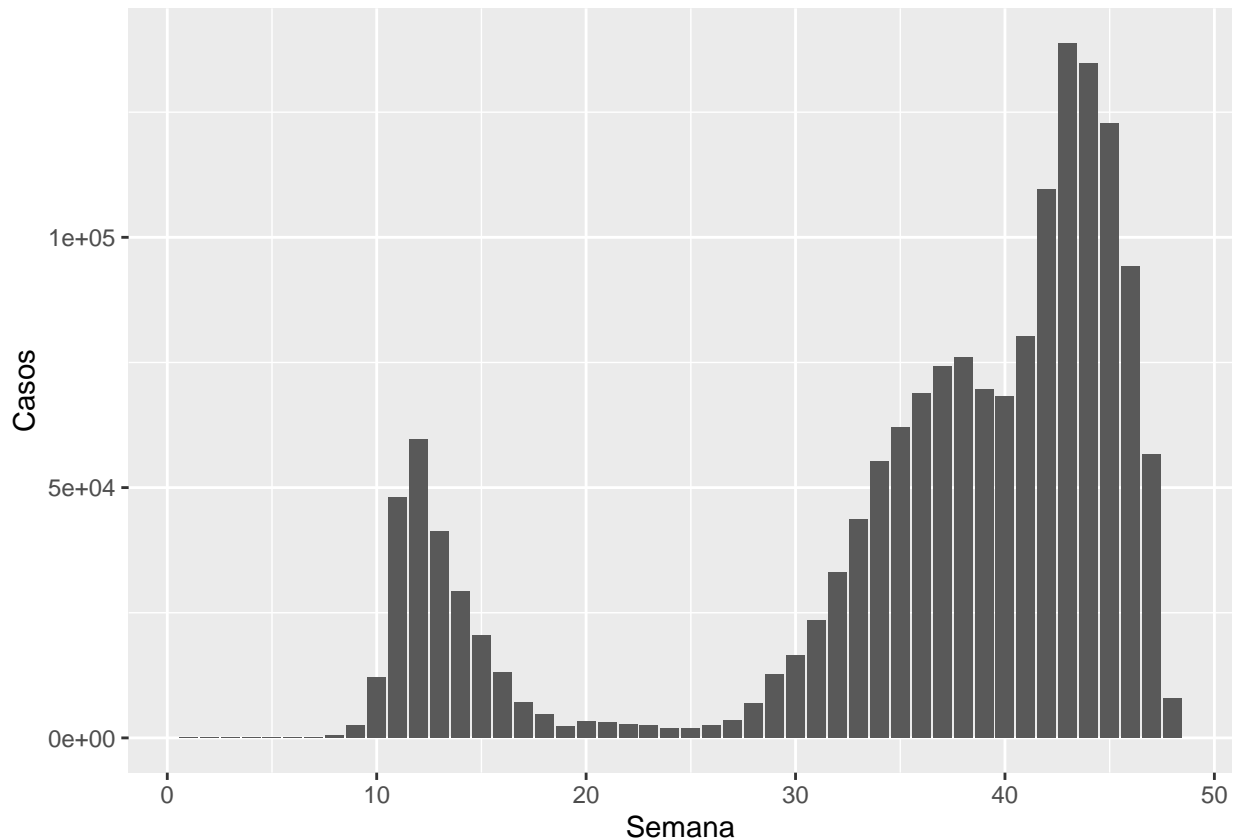
El número total de casos acumulados desde el 1 de enero de 2020 hasta la fecha indicada en el punto anterior según los datos oficiales disponibles en ese momento ascienden a un total de **1.617.504**.

Considerando una población en España de **47,33** 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 **3,418 %** hasta la fecha.

## Incidencia semanal

- Evolución de número de casos identificados por semanas:

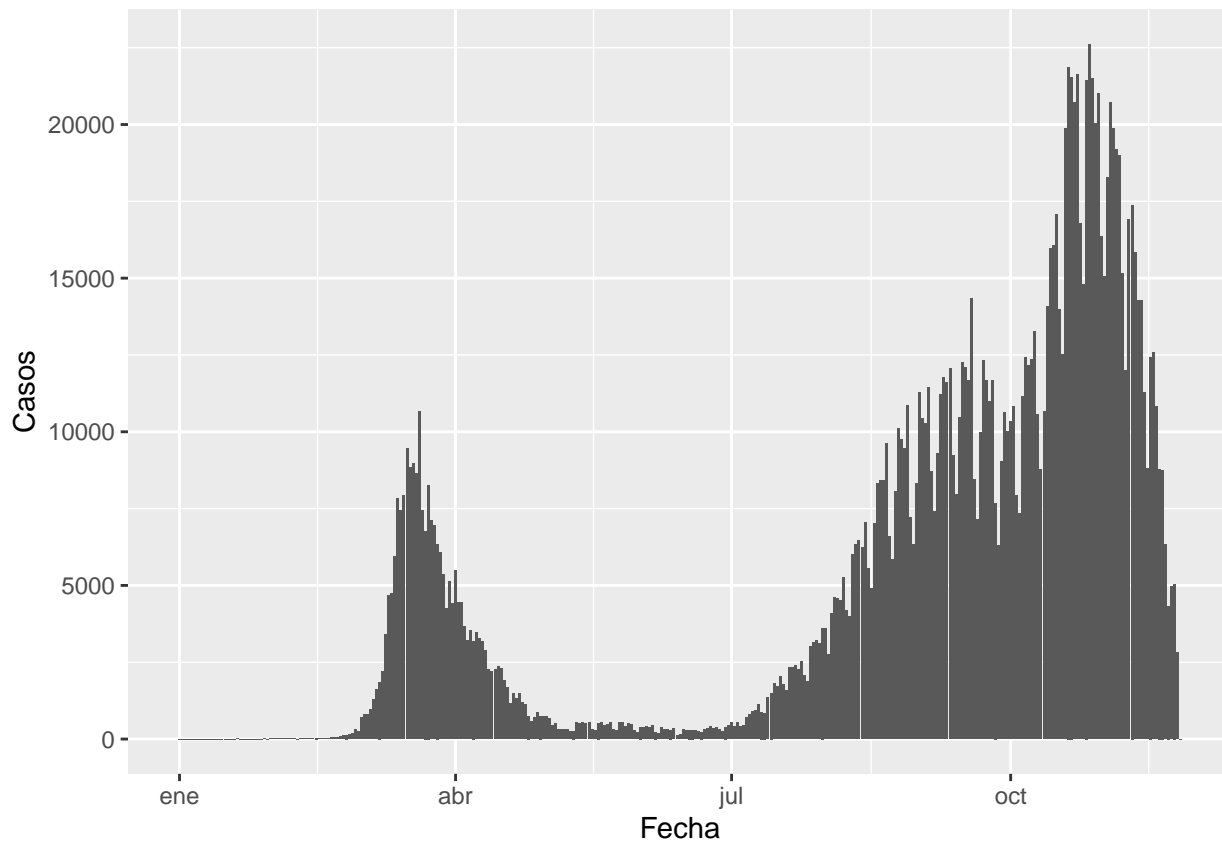
```
CasosSemanales <- tapply(Datos$num_casos, week(Datos$fecha), sum)
dfCasosSemanales <- data.frame(Semana = 1:length(CasosSemanales), Casos =
                                CasosSemanales)
d <- ggplot(dfCasosSemanales, aes(Semana, Casos))
d + geom_col()
```



## Incidencia diaria

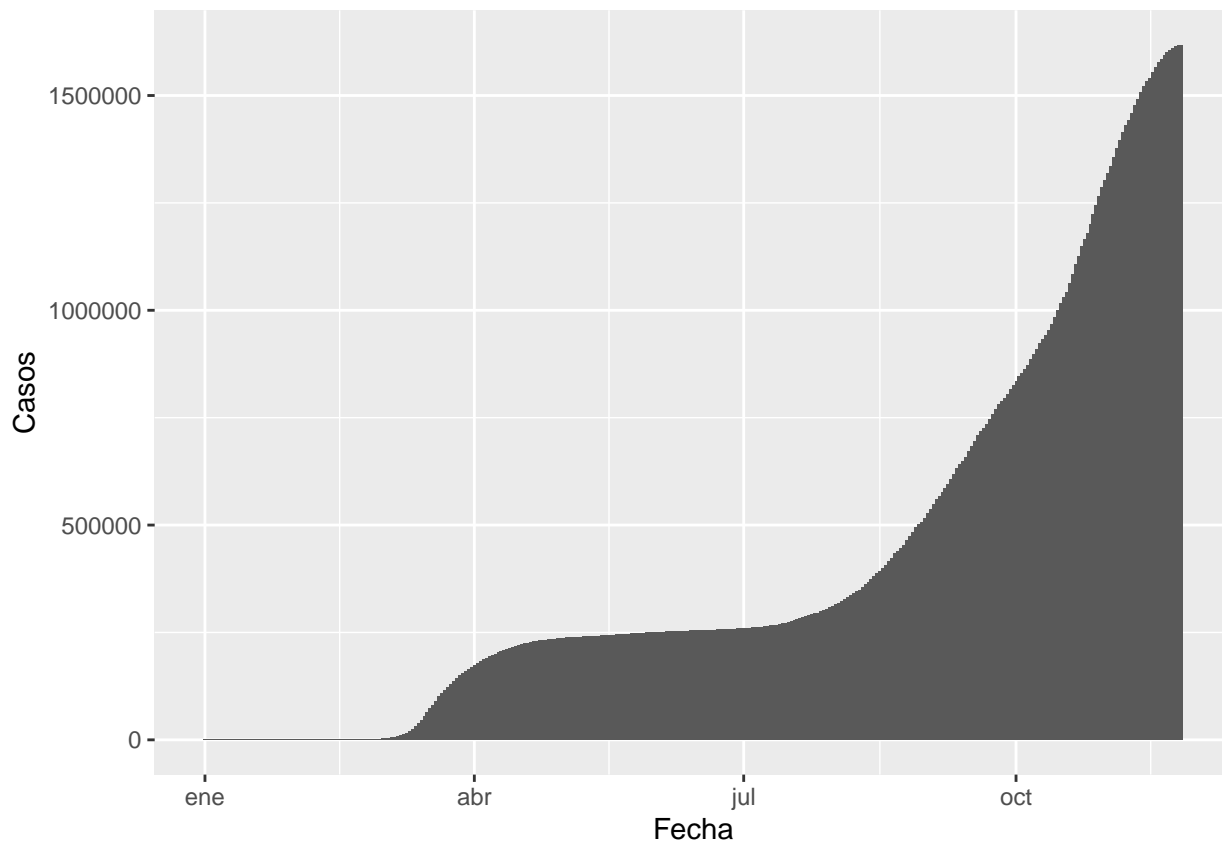
- Curva epidémica de los casos notificados por días:

```
dfCasosDiarios <- data.frame(Fecha = Datos$fecha, Casos = Datos$num_casos)
d <- ggplot(dfCasosDiarios, aes(Fecha, Casos))
d + geom_col()
```



- Gráfico de casos acumulados a origen por días:

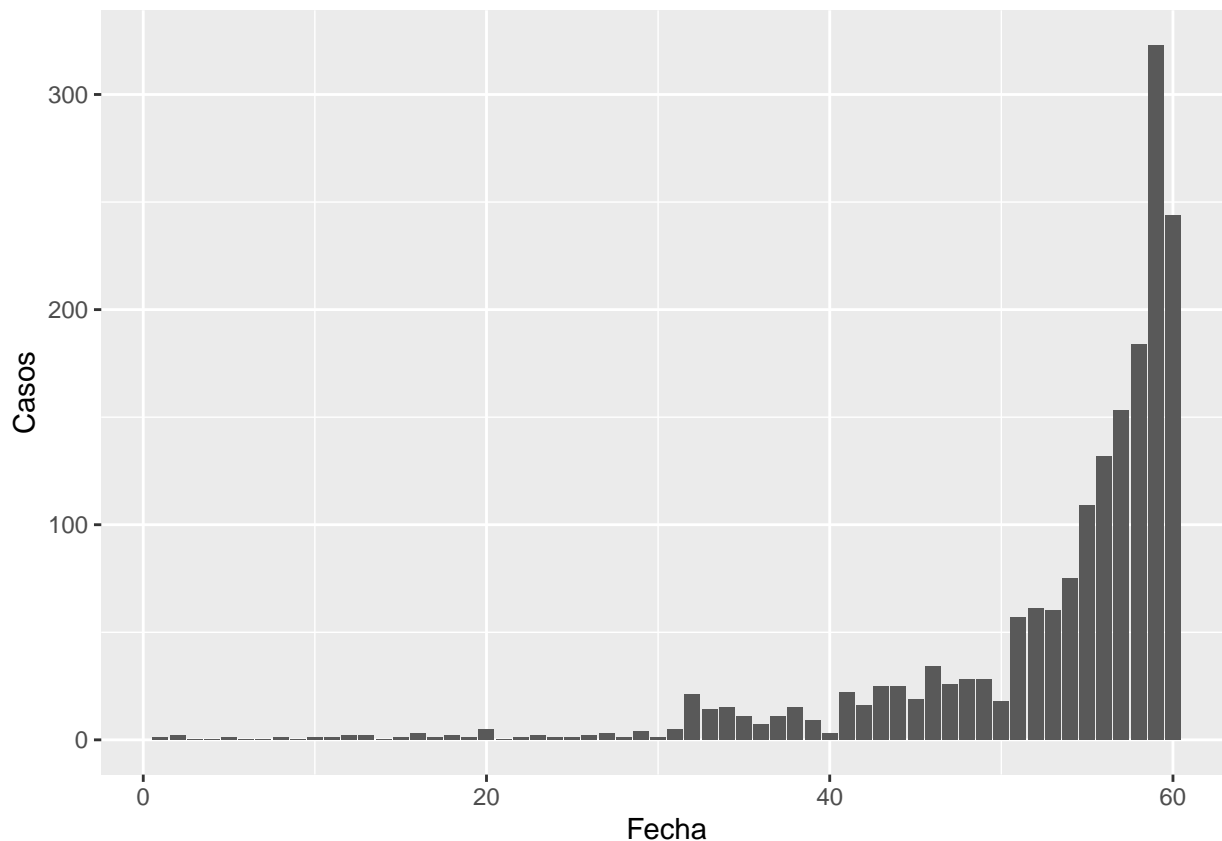
```
CasosDiariosAcumul <- cumsum(dfCasosDiarios$Casos)
dfCasosDiariosAcumul <- data.frame(Fecha = Datos$fecha, Casos = CasosDiariosAcumul)
d <- ggplot(dfCasosDiariosAcumul, aes(Fecha, Casos))
d + geom_col()
```



## Detalle del número de casos en los dos 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(Fecha = 1:length(CasosDiariosEF), Casos = CasosDiariosEF)
TotalCasosEF <- sum(DatosEF$num_casos)
d <- ggplot(dfCasosDiariosEF, aes(Fecha, Casos))
d + geom_col()
```



Los casos reportados totales a lo largo de esos dos meses son **1.790**, 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: 392
- Periodo 51-60: 1.398

En 10 días se detectan **3,6** 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 **10.186** casos a añadir al total anterior, siendo éstos **5,7** veces los registrados a lo largo de todo enero y febrero.

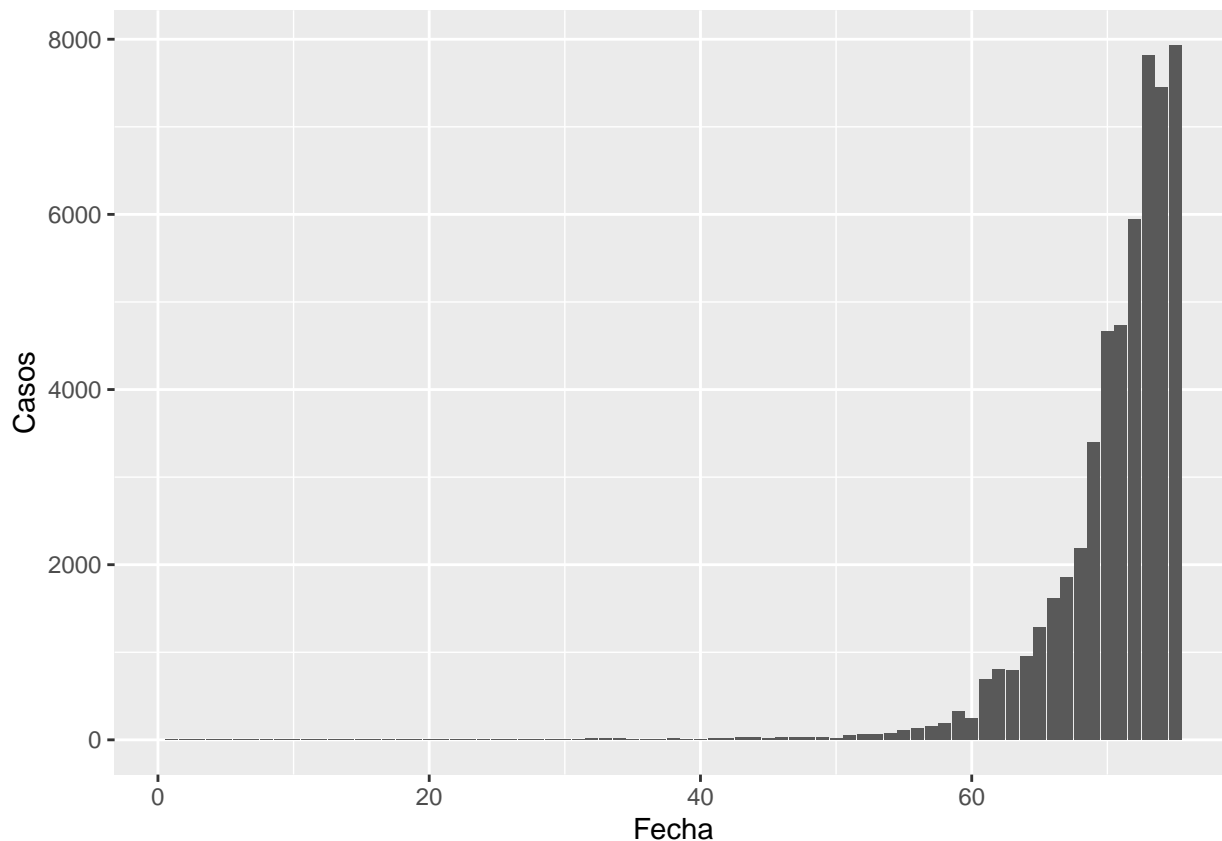
Durante los siguientes siete días, del 9 al 15 de marzo, los casos a sumar fueron **41.939**, lo que supone **4,1** veces los notificados en los 8 primeros días del mes.

- 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 <- ggplot(dfCasosDiariosHasta15mar, aes(Fecha, Casos))
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 la segunda ola de infecciones, donde se ha hablado de 200, 500 e incluso 1.000 casos/100.000 hab.

## Evolución de la incidencia acumulada a lo largo de todo el año

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=integer())
IA14aux <- as.integer()
InicioPeriodo7 <- as.Date("2020-01-01")
FinPeriodo7 <- InicioPeriodo7 + 6
IncidAcum7 <- data.frame(Fecha=as.Date(character()), IA7=integer())
IA7aux <- as.integer()

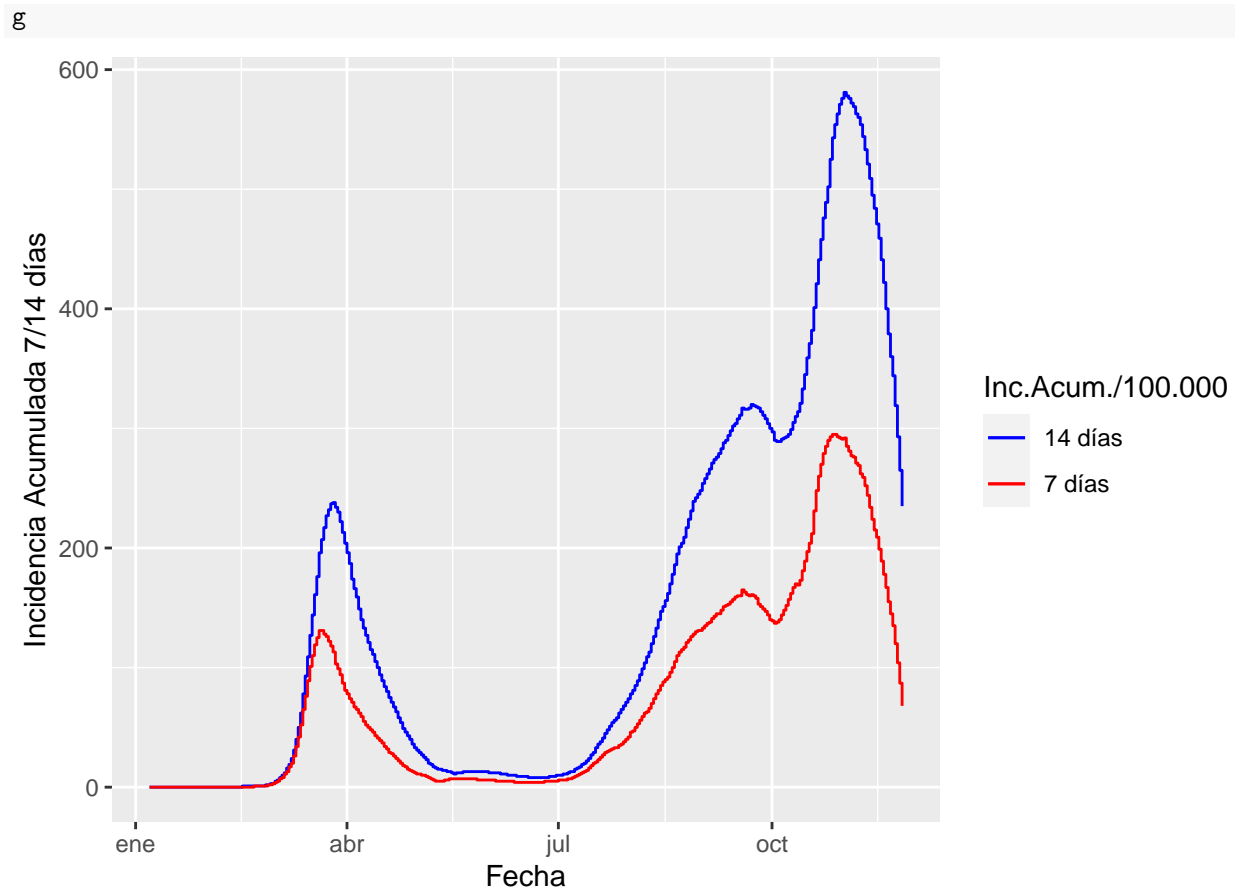
## Incidencia acumulada 14 días
while (FinPeriodo14 <= UltFecha) {
  IA14aux <- 0
  Periodo14 <- seq.Date(
    from = InicioPeriodo14,
    to = FinPeriodo14,
    by = "day")
  DatosIA14 <- subset(Datos, fecha %in% Periodo14)
  IA14aux <- round(sum(DatosIA14$num_casos)/poblESP*1E+05, digits = 0)
  IncidAcum14 <- add_row(IncidAcum14, Fecha = FinPeriodo14, IA14 = IA14aux)
  InicioPeriodo14 <- InicioPeriodo14 + 1
  FinPeriodo14 <- FinPeriodo14 + 1
}

## Incidencia acumulada 7 días
while (FinPeriodo7 <= UltFecha) {
  IA7aux <- 0
  Periodo7 <- seq.Date(
    from = InicioPeriodo7,
    to = FinPeriodo7,
    by = "day")
  DatosIA7 <- subset(Datos, fecha %in% Periodo7)
  IA7aux <- round(sum(DatosIA7$num_casos)/poblESP*1E+05, digits = 0)
  IncidAcum7 <- add_row(IncidAcum7, Fecha = FinPeriodo7, IA7 = IA7aux)
  InicioPeriodo7 <- InicioPeriodo7 + 1
  FinPeriodo7 <- FinPeriodo7 + 1
}

## Gráficos de Incidencia acumulada por separado
## d <- ggplot(IncidAcum14, aes(Fecha, IA14)) + labs(y = "Incidencia Acumulada 14 días")
## d + geom_col()
## e <- ggplot(IncidAcum7, aes(Fecha, IA7)) + labs(y = "Incidencia Acumulada 7 días")
## e + geom_col()

## 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 <- ggplot(IncidAcum7_14, aes(Fecha)) + labs(y = "Incidencia Acumulada 7/14 días")
g <- g + scale_color_manual(name="Inc.Acum./100.000",
  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"))

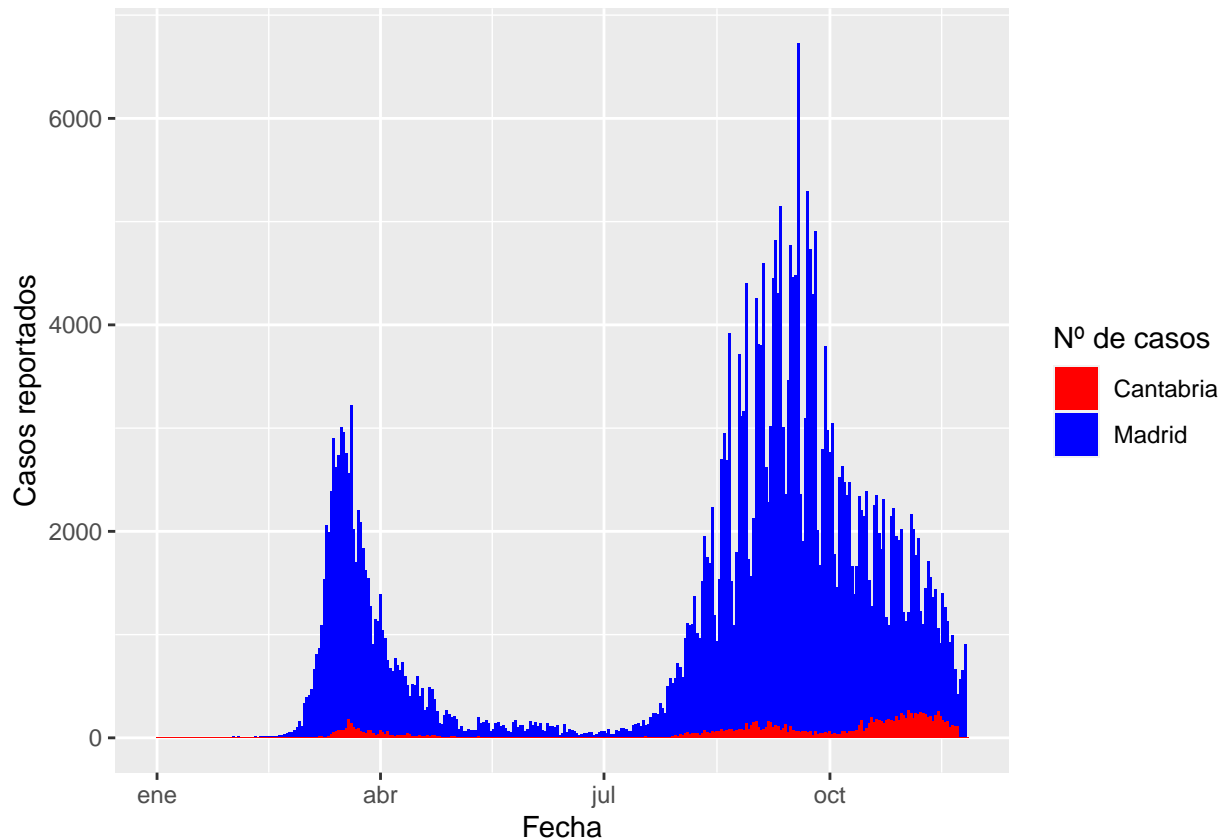
```



## Comparación de la evolución del número de casos entre Cantabria y Madrid

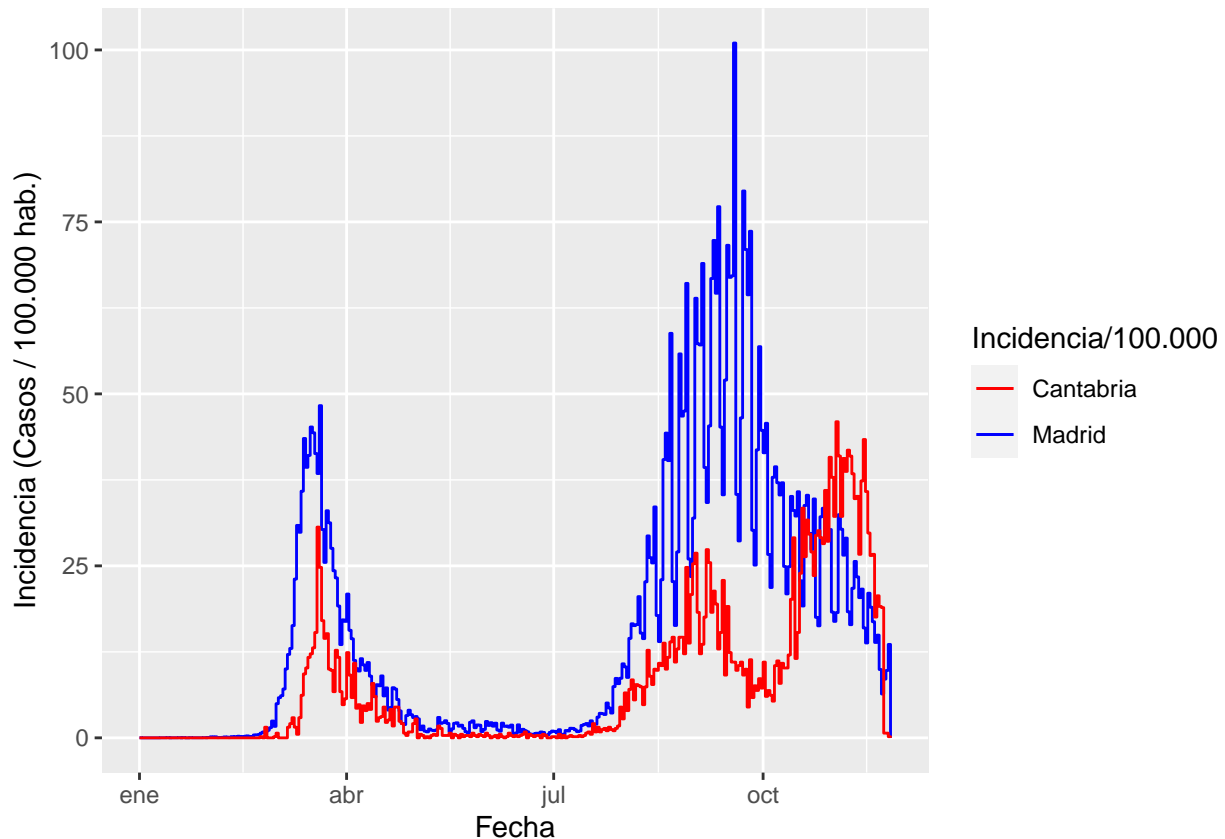
En el siguiente gráfico se compara la evolución de la enfermedad entre dos comunidades muy diferentes, Cantabria y la Comunidad Autónoma de Madrid:

```
## ISO CCAA: Cantabria = CB, Madrid = MD
CasosDiariosCantabria <- filter(DatosCCAA, ccaa_iso == "CB")
CasosDiariosCantabria <- subset (CasosDiariosCantabria, select = -ccaa_iso)
colnames(CasosDiariosCantabria) <- c("Fecha", "Casos")
CasosDiariosMadrid <- filter(DatosCCAA, ccaa_iso == "MD")
CasosDiariosMadrid <- subset (CasosDiariosMadrid, select = -ccaa_iso)
colnames(CasosDiariosMadrid) <- c("Fecha", "Casos")
CasosCantabriaMadrid <- add_column(CasosDiariosCantabria, CasosDiariosMadrid$Casos)
colnames(CasosCantabriaMadrid) <- c("Fecha", "CasosCANT", "CasosMAD")
g <- ggplot(CasosCantabriaMadrid, aes(Fecha)) + labs(y = "Casos reportados")
g <- g + scale_fill_manual(name="Nº de casos",
                           values = c("Madrid" = "blue", "Cantabria" = "red"))
g <- g + geom_col(aes(y=CasosMAD, fill="Madrid"))
g <- g + geom_col(aes(y=CasosCANT, fill="Cantabria"))
g
```



Como es lógico, los datos no son comparables en términos absolutos por la gran diferencia de población. Replantamos ahora número de casos por cada 100.000 habitantes, con los datos de población en cada comunidad disponibles en el momento en el INE, que corresponden a 2019, reflejando los datos de Cantabria en color rojo y los de Madrid en azul:

```
poblMAD <- 6.663E+06
poblCANT <- 581E+03
IncidenciaCantabriaMadrid <- data.frame(CasosCantabriaMadrid$Fecha,
                                         CasosDiariosCantabria$Casos/poblCANT*1E+05,
                                         CasosDiariosMadrid$Casos/poblMAD*1E+05)
colnames(IncidenciaCantabriaMadrid) <- c("Fecha", "IncidCANT", "IncidMAD")
g <- ggplot(IncidenciaCantabriaMadrid, aes(Fecha)) +
  labs(y = "Incidencia (Casos / 100.000 hab.)")
g <- g + scale_color_manual(name="Incidencia/100.000",
                           values = c("Madrid" = "blue", "Cantabria" = "red"))
g <- g + geom_step(aes(y=IncidMAD, colour="Madrid"))
g <- g + geom_step(aes(y=IncidCANT, colour="Cantabria"))
## g <- g + theme(legend.justification = "center", legend.position = "right")
g
```



Por completar la información comparativa entre ambas comunidades se adjunta también la incidencia acumulada en 14 días para ambas áreas geográficas, junto con la correspondiente al conjunto del territorio nacional:

```
InicioPeriodo14 <- as.Date("2020-01-01")
FinPeriodo14 <- InicioPeriodo14 + 13
IA14_CantabriaMadrid <- data.frame(Fecha=as.Date(character()), IA14_CANT=integer(),
                                   IA14_MAD=integer())

IA14aux <- as.integer()
while (FinPeriodo14 <= UltFecha) {
  IA14auxCANT <- 0
  IA14auxMAD <- 0
  Periodo14 <- seq.Date(
    from = InicioPeriodo14,
    to = FinPeriodo14,
    by = "day")

  DatosIA14_CantMad <- subset(IncendenciaCantabriaMadrid, Fecha %in% Periodo14)
  IA14auxCANT <- round(sum(DatosIA14_CantMad$IncidCANT), digits = 0)
  IA14auxMAD <- round(sum(DatosIA14_CantMad$IncidMAD), digits = 0)
  IA14_CantabriaMadrid <- add_row(IA14_CantabriaMadrid, Fecha = FinPeriodo14,
                                IA14_CANT = IA14auxCANT, IA14_MAD = IA14auxMAD)

  InicioPeriodo14 <- InicioPeriodo14 + 1
  FinPeriodo14 <- FinPeriodo14 + 1
}

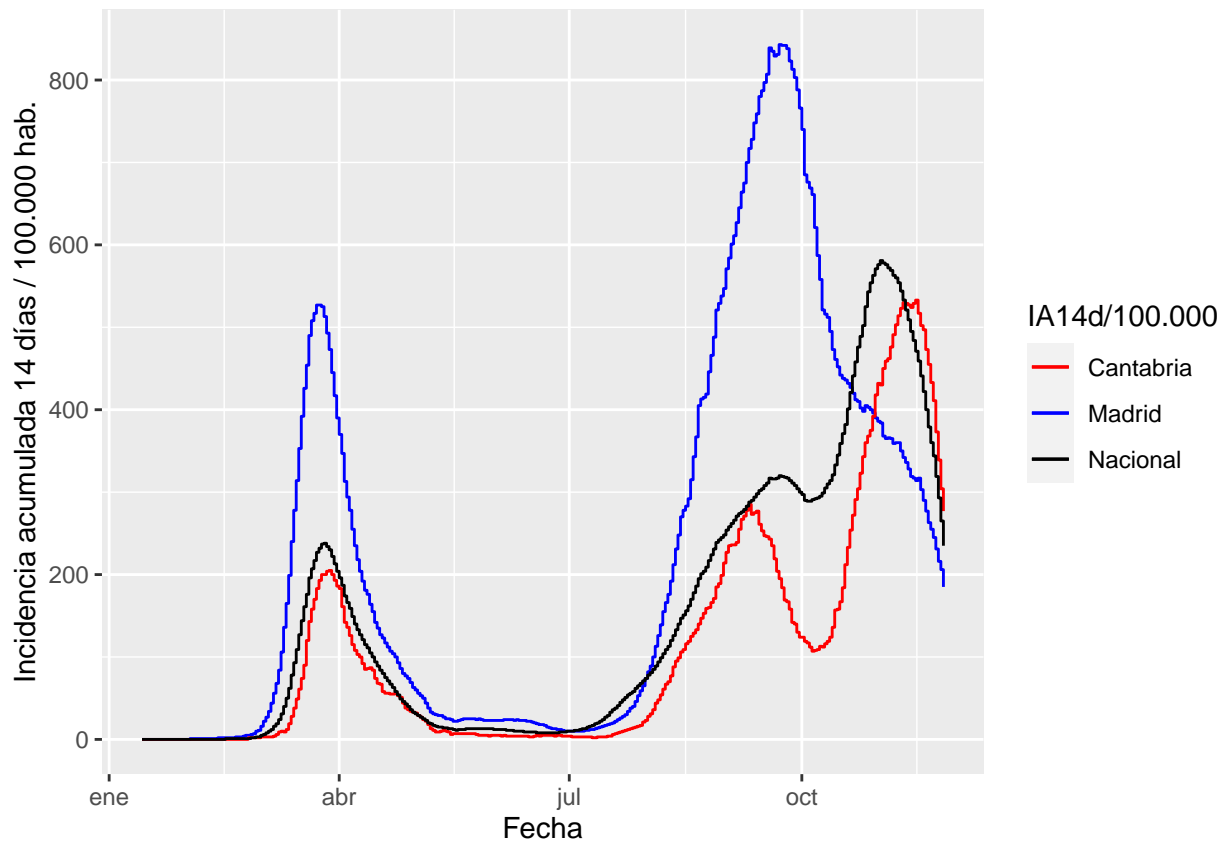
## Añadiendo columna con valores a nivel nacional
IA14_CantMadNac <- add_column(IA14_CantabriaMadrid, IA14_Nacional=IncidAcum14$IA14)
## Gráfico
g <- ggplot(IA14_CantMadNac, aes(Fecha))
```

```

g <- g + labs(y = "Incidencia acumulada 14 días / 100.000 hab.")
g <- g + scale_color_manual(name = "IA14d/100.000",
                             values = c("Madrid" = "blue", "Cantabria" =
                                           "red", "Nacional" = "black"))

g <- g + geom_step(aes(y=IA14_MAD, color="Madrid"))
g <- g + geom_step(aes(y=IA14_CANT, color="Cantabria"))
g <- g + geom_step(aes(y=IA14_Nacional, color="Nacional"))
## g <- g + theme(legend.justification = "center", legend.position = "right")
g

```



## Exceso de mortalidad

Como último paso del análisis obtendremos cifras del exceso de mortalidad registrado en este año, presumiblemente achacable a la incidencia de la pandemia de la COVID-19. Los datos se han obtenido del enlace del **Instituto de Salud Carlos III**: <https://momo.isciii.es/public/momo/data>.

```

DatosCompletosMoMo <- read.csv(file.path("data", "data.csv"))
## Descartamos información de detalle de CCAA, sexos y grupos de edad
DatosMoMo <- DatosCompletosMoMo[DatosCompletosMoMo$ambito == "nacional" &
                                 DatosCompletosMoMo$cod_sexo == "all" &
                                 DatosCompletosMoMo$cod_gedad == "all" &
                                 DatosCompletosMoMo$fecha_defuncion > "2019-12-31",
                                 c(9:10, 13:15)]

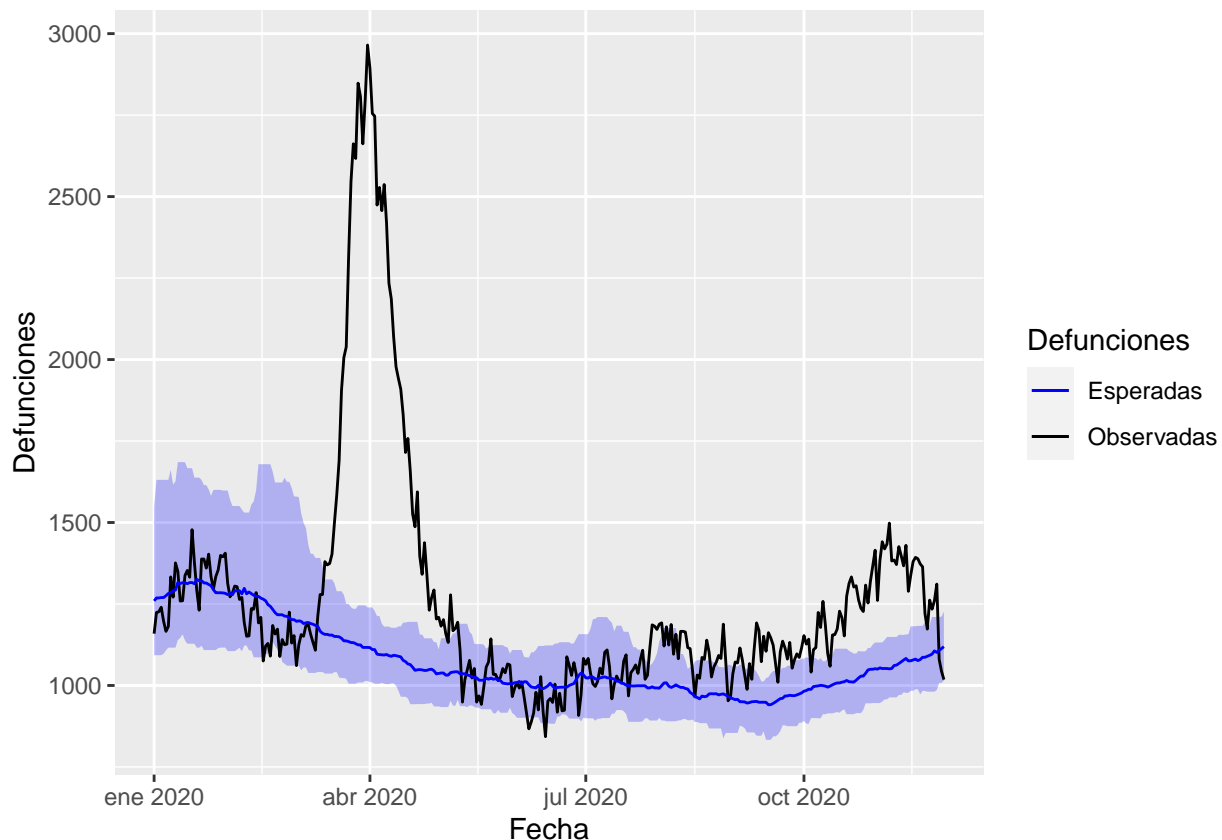
colnames(DatosMoMo) <- c("Fecha", "DefObs", "DefEsp", "DefEsp_q01", "DefEsp_q99")
DatosMoMo <- as.data.table(DatosMoMo)
## Adecuación formato fechas para su correcta lectura
DatosMoMo$Fecha <- as.Date(DatosMoMo$Fecha, format = "%Y-%m-%d")

```

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): **2020-11-30 15:13:41**

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 <- ggplot(DatosMoMo, aes(Fecha))
g <- g + labs(y = "Defunciones")
g <- g + scale_color_manual(name = "Defunciones",
                           values = c("Observadas"="black", "Esperadas"="blue"))
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 + theme(legend.justification = "center", legend.position = "right")
g
```



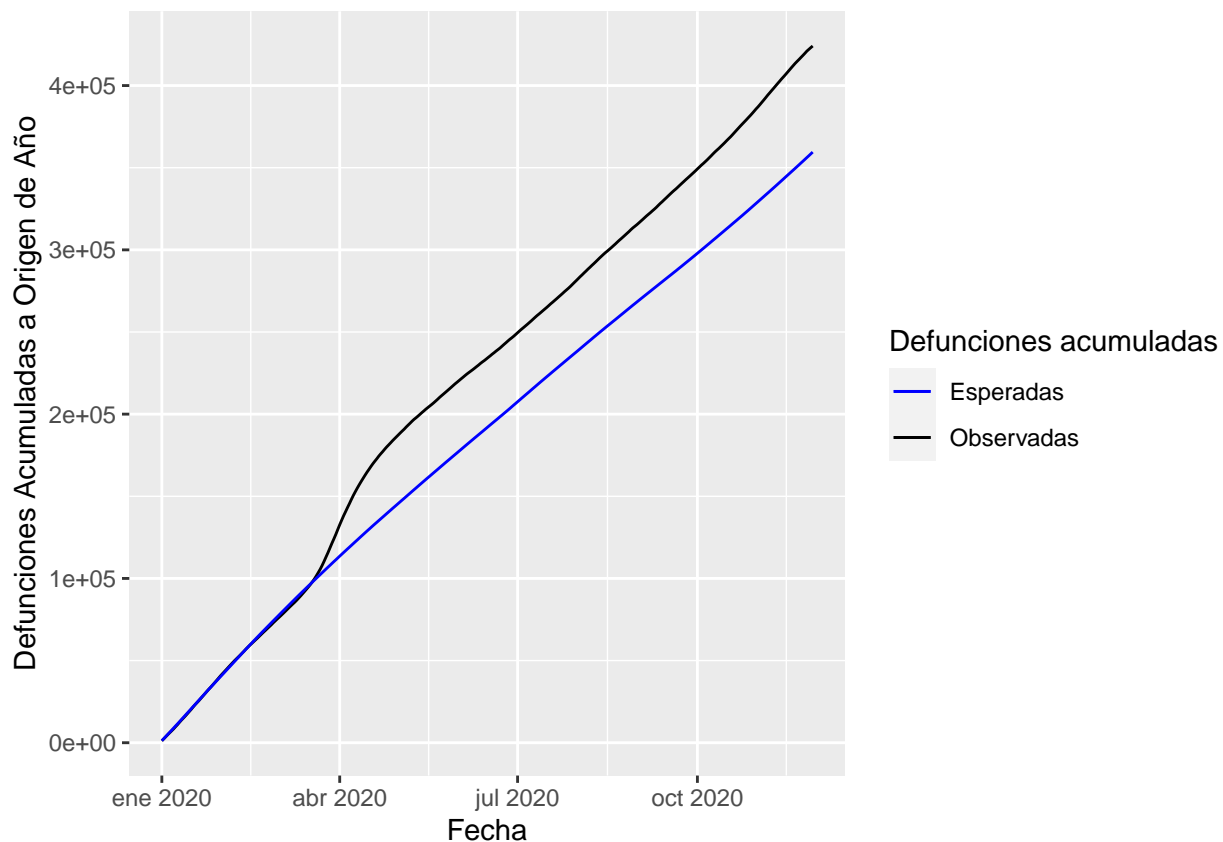
Como se puede ver, existe un periodo de mortalidad totalmente disparada a lo largo de los meses de marzo a mayo, mientras que luego se aprecia otro periodo de desviación al alza, no tan acusado pero más prolongado en el tiempo y con una tendencia creciente, que cubriría desde agosto hasta bien avanzado octubre.

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 del año con las que realmente se han registrado:

```
DatosMoMo[,AcumObs := cumsum(DefObs)]
DatosMoMo[,AcumEsp := cumsum(DefEsp)]
g <- ggplot(DatosMoMo, aes(Fecha))
g <- g + labs(y = "Defunciones Acumuladas a Origen de Año")
g <- g + scale_color_manual(name = "Defunciones acumuladas",
                           values = c("Observadas"="black", "Esperadas"="blue"))
g <- g + geom_line(aes(y=AcumObs, color="Observadas"))
g <- g + geom_line(aes(y=AcumEsp, color="Esperadas"))
## g <- g + theme(legend.justification = "center", legend.position = "right")
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 despegan de las esperadas de forma muy visible a lo largo de los meses de marzo a mayo, y que dicho distanciamiento se vuelve a incrementar, ya a menor ritmo, a partir del mes de agosto, aunque vuelve a repuntar en noviembre.

```
## 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, una total de **424.094** defunciones observadas y **359.518** defunciones esperadas, resultando un exceso de **64.576** defunciones. Expresando dicho exceso en términos porcentuales, nos encontramos con un **18 %** 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 la 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: **197.398** personas
- Exceso de defunciones: **41.911** personas
- En tanto por ciento: **21,2 %**

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:

```
## Columnas auxiliares variables booleanas
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$Fecha[i+k])
        PeriodosExceso <- add_row(PeriodosExceso,
                                   Inicio = PeriodosExcesoInicioN,
                                   Fin = PeriodosExcesoFinN)
        break
      }
      if (DatosMoMo$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 + k
        break
      }
    }
    i <- i + 1
  }
}
## Unir periodos de exceso próximos
```



```

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
2020-08-31	2020-11-26

- Después de unificar los periodos de exceso cercanos ( $\leq 2$  días):

Inicio	Fin
2020-03-10	2020-05-09
2020-07-20	2020-11-26

```

## Excesos de defunciones durante los periodos de exceso
PeriodoExcesoN <- as.Date(character())
ExcesoDefunciones <- data.table("Exceso"=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))))
}
PeriodosExcesoUnido <- add_column(PeriodosExcesoUnido, "Exceso de defunciones" =
  ExcesoDefunciones$Exceso)
TotalExcesoDefunciones <- sum(PeriodosExcesoUnido$"Exceso de defunciones")

```

Los excesos de defunciones en estos 2 periodos son:

Inicio	Fin	Exceso de defunciones
2020-03-10	2020-05-09	44.599
2020-07-20	2020-11-26	22.443

Siendo el total agregado de exceso de defunciones de **67.042** personas.

Expresándolo en términos porcentuales, el exceso de defunciones es un **18,6 %** 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.

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 provocado por la pandemia, los fallecimientos asociados a las mismas son por tanto atribuibles a la COVID-19 aunque el virus no haya tenido presencia en el paciente correspondiente.

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 ya que el propio estado de alarma tuvo por necesidad incidencia negativa en el número de fallecimientos por accidente laboral y por accidente de tráfico, teniendo que rebajarse por tanto el patrón de referencia de defunciones durante el estado de confinamiento y arrojando un exceso de defunciones por causa de la COVID-19 superiores a los mostrados más arriba. Aunque es posible realizar estimaciones de estas desviaciones con datos disponibles públicamente, dejamos esa posibilidad de perfeccionamiento del estudio para mejor oportunidad.

.....

## Referencias

- (1) 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/>
- (2) 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/>
- (3) Yihui Xie (2020). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.30.
- (4) Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, <https://doi.org/10.21105/joss.01686>
- (5) 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>