

# Disease Mapping

Zaira García, Luis Requeni y Celia Sifre

12/12/2022

El principal objetivo de este estudio es valorar la mortalidad por enfermedad isquémica en hombres en la comunidad autónoma de Aragón durante el periodo 1991-2000. Para ello, realizaremos un mapeo de la isquemia en dicha comunidad. Disponemos de una base de datos que contiene el número de casos esperados y observados por municipio.

En primer lugar, crearemos la estructura de directorios inicial del proyecto que facilite su reproducibilidad para la persona interesada.

```
dir.create("r")
dir.create("figuras")
dir.create("informes")
dir.create("datos")
dir.create("datos/brutos")
dir.create("datos/procesados")
```

A continuación, cargaremos los datos y definiremos la relación de vecindad entre los distintos municipios. En concreto, la relación de vecindad que escogeremos será por contigüidad.

```
load(file.path("../datos/procesados", "Aragon.Rdata"))
aragon.shp <- readOGR(file.path("../datos/procesados", "aragon.shp"))
aragon.shp <- aragon.shp[order(aragon.shp$CODMUNI), ]
aragon.nb <- poly2nb(aragon.shp)
vecinos <- nb2WB(aragon.nb)
paleta <- colorRampPalette(brewer.pal(9, 'Blues'))(5)
```

Nuestra variable respuesta  $O_i$  representa los casos observados por enfermedad isquémica en hombres. Por tanto, tenemos una variable cuantitativa discreta y se distribuye como una Poisson de parámetro  $\mu$ :

$$O_i \sim \text{Po}(\mu_i).$$

Para realizar dicha modelización, utilizaremos la propuesta de Besag, York y Mollié para el análisis espacial de riesgos en áreas pequeñas. Implementaremos este modelo tanto en WinBUGS como en INLA.

- WinBUGS

```

modelo <- function()
{
  for(i in 1:n){
    O[i] ~ dpois(mu[i])
    log(mu[i]) <- log(E[i]) + m + het[i] + sp[i]
    het[i] ~ dnorm(0, prechet)
    R[i] <- exp(m + het[i] + sp[i])
  }
  sp[1:n] ~ car.normal(adj[], w[], num[], precsp)
  m ~ dflat()
  prechet <- pow(sdhet, -2)
  precsp <- pow(sdsp, -2)
  sdhet ~ dunif(0, 10)
  sdsp ~ dunif(0, 10)
}

```

```

set.seed(123)

datos <- list(E = Aragon.df$E, O = Aragon.df$O, n = dim(Aragon.df)[1],
              adj = vecinos$adj, w = vecinos$weights, num = vecinos$num)
parametros <- c("mu", "sdhet", "sdsp", "m", "R")
iniciales <- function()
{
  list(sdhet = runif(1), sdsp = runif(1), m = rnorm(1))
}
iteraciones <- 20000
burnin <- 2000

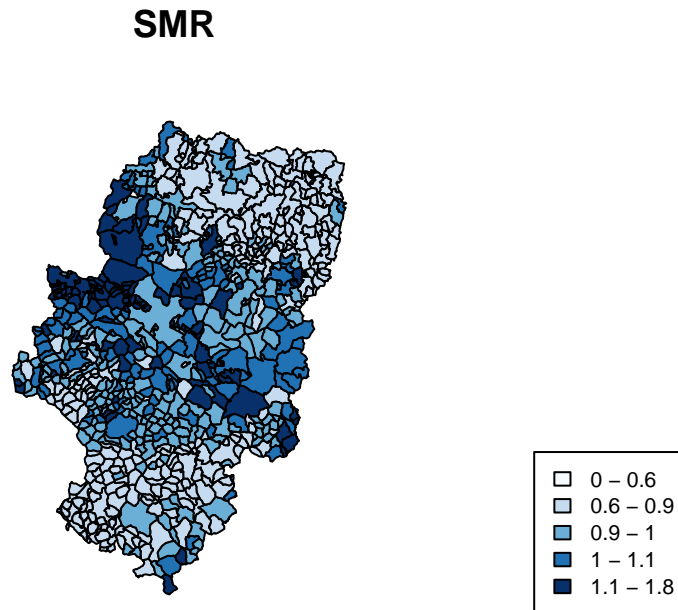
ajuste.modelo <- bugs(model = modelo, inits = iniciales, data = datos,
                     parameters.to.save = parametros,
                     n.iter = iteraciones, n.burnin = burnin)
save(ajuste.modelo, file = "ajuste.modelo.rda")

```

Hemos realizado un total de 20000 iteraciones y eliminamos las 2000 primeras para asegurar la convergencia en dicha simulación.

Representaremos el SMR de la isquemia en hombres estimado por nuestro modelo.

```
plot(aragon.shp, col = paleta[findInterval(ajuste.modelo$mean$R,
                                           c(0, 0.6, 0.9, 1, 1.1, 1.8))])
title("SMR ")
legend("bottomright", c("0 - 0.6", "0.6 - 0.9", "0.9 - 1", "1 - 1.1", "1.1 - 1.8"),
      fill = paleta, cex = 0.7)
```



- INLA

```
temp <- poly2nb(aragon.shp)
nb2INLA(file.path("../figuras", "LDN.graph"), temp)

H <- inla.read.graph(filename = file.path("../figuras", "LDN.graph"))
S <- U <- seq(1, 729)
data <- cbind(Aragon.df, S, U)
formula <- 0 ~ 1 + f(S,
                     model = "besag",
                     graph = H,
                     scale.model = TRUE,
                     hyper = list(prec = list(prior = "loggamma", param = c(1, 0.001)))) +
  f(U,
    model = "iid",
    hyper = list(prec = list(prior = "loggamma", param = c(1, 0.001))))
```

```

modelo.inla <- inla(formula,
  family = "poisson",
  data = Aragon.df,
  E = E,
  control.compute = list(dic = TRUE, waic = TRUE, cpo = TRUE),
  control.predictor = list(compute = TRUE, cdf = c(log(1))))
summary(modelo.inla)

```

```

##
## Call:
##   c("inla.core(formula = formula, family = family, contrasts = contrasts,
##   ", " data = data, quantiles = quantiles, E = E, offset = offset, ", "
##   scale = scale, weights = weights, Ntrials = Ntrials, strata = strata,
##   ", " lp.scale = lp.scale, link.covariates = link.covariates, verbose =
##   verbose, ", " lincomb = lincomb, selection = selection, control.compute
##   = control.compute, ", " control.predictor = control.predictor,
##   control.family = control.family, ", " control.inla = control.inla,
##   control.fixed = control.fixed, ", " control.mode = control.mode,
##   control.expert = control.expert, ", " control.hazard = control.hazard,
##   control.lincomb = control.lincomb, ", " control.update =
##   control.update, control.lp.scale = control.lp.scale, ", "
##   control.pardiso = control.pardiso, only.hyperparam = only.hyperparam,
##   ", " inla.call = inla.call, inla.arg = inla.arg, num.threads =
##   num.threads, ", " blas.num.threads = blas.num.threads, keep = keep,
##   working.directory = working.directory, ", " silent = silent, inla.mode
##   = inla.mode, safe = FALSE, debug = debug, ", " .parent.frame =
##   .parent.frame)")
## Time used:
##   Pre = 3.53, Running = 5.81, Post = 0.0413, Total = 9.38
## Fixed effects:
##           mean      sd 0.025quant 0.5quant 0.975quant mode kld
## (Intercept) -0.061 0.036      -0.134      -0.06       0.007  NA   0
##
## Random effects:
##   Name      Model
##   S Besags ICAR model
##   U IID model
##
## Model hyperparameters:
##           mean      sd 0.025quant 0.5quant 0.975quant mode
## Precision for S  21.69 14.11       7.23   17.83      59.29  NA
## Precision for U 486.98 650.79       3.75  236.16     2278.49  NA
##
## Deviance Information Criterion (DIC) .....: 1899.64
## Deviance Information Criterion (DIC, saturated) ....: -5209.42
## Effective number of parameters .....: 50.67
##
## Watanabe-Akaike information criterion (WAIC) ...: 1904.36
## Effective number of parameters .....: 49.89
##
## Marginal log-Likelihood: -1246.49
## CPO, PIT is computed
## Posterior summaries for the linear predictor and the fitted values are computed

```

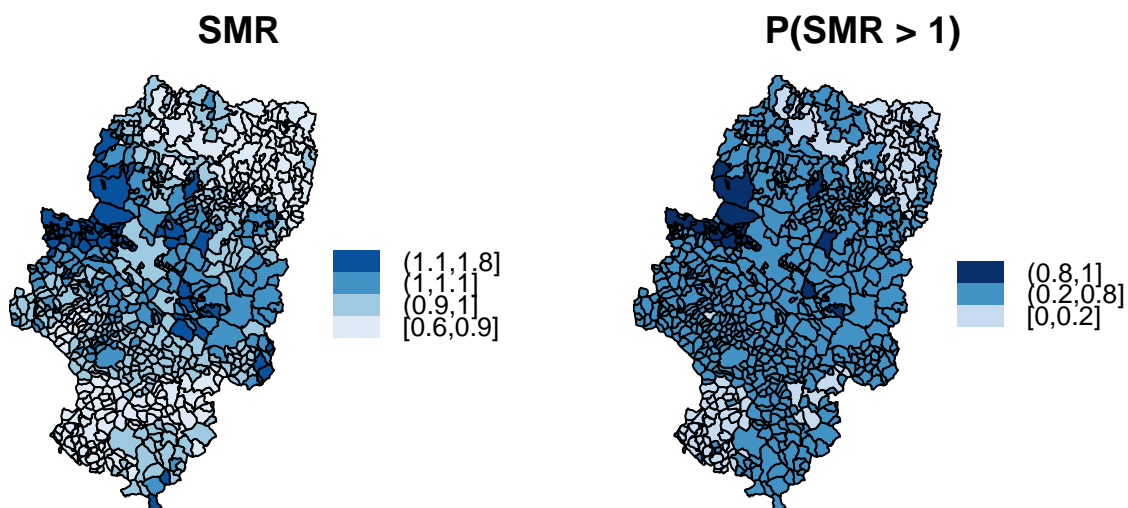
```
## (Posterior marginals needs also 'control.compute=list(return.marginals.predictor=TRUE)')
```

Representaremos tanto el SMR como la probabilidad de que el SMR sea mayor que 1 de la isquemia en hombres estimado por nuestro modelo.

```
# media
aragon.shp$SMR_mean <- modelo.inla$summary.fitted.values$mean
# desviación típica
aragon.shp$SMR_sd <- modelo.inla$summary.fitted.values$sd
# probabilidad de ser mayor que 1
aragon.shp$SMR_p1 <- 1 - modelo.inla$summary.fitted.values$`1 cdf`

SMR.cutoff <- c(0.6, 0.9, 1.0, 1.1, 1.8)
SMR_p1.cutoff <- c(0, 0.2, 0.8, 1)
SMR_disc <- cut(aragon.shp$SMR_mean,
                breaks = SMR.cutoff,
                include.lowest = TRUE)
SMR_p1_disc <- cut(aragon.shp$SMR_p1,
                  breaks = SMR_p1.cutoff,
                  include.lowest = TRUE)

aragon.shp$SMR_disc <- SMR_disc
aragon.shp$SMR_p1_disc <- SMR_p1_disc
grid.arrange(spplot(aragon.shp,
                    c("SMR_disc"),
                    col.regions = brewer.pal(9, 'Blues')[c(2, 4, 6, 8)],
                    main = "SMR",
                    par.settings = list(axis.line = list(col = 'transparent'))),
            spplot(aragon.shp,
                    c("SMR_p1_disc"),
                    col.regions = brewer.pal(9, 'Blues')[c(3, 6, 9)],
                    main = "P(SMR > 1)",
                    par.settings = list(axis.line = list(col = 'transparent'))), ncol = 2)
```



## Conclusiones

A la vista de los resultados ofrecidos por ambas técnicas, podemos concluir que parece existir un mayor SMR en los municipios cercanos al río Ebro. Esto nos hace pensar en la presencia de algún factor en esas zonas que influya en los casos de isquemia. Aunque sin fundamento objetivo alguno, podríamos apostar a que la alta actividad industrial de la zona puede ser la responsable de ello.

Para facilitar su reproducibilidad, hemos creado un proyecto en R incluyendo la creación de los distintos archivos que facilite la organización de la información. El informe se ha desarrollado a través de un fichero de RmarkDown permitiendo al mismo tiempo la programación y ejecución de comandos a través de R así como la edición del texto en un formato amigable.

En este sentido utilizamos igualmente rutas de archivos relativas que permiten que cualquier persona con su ordenador y abriendo el proyecto puede utilizar los distintos archivos y ficheros de manera sencilla. Igualmente, hacemos uso de la función *file.path()* que soluciona los problemas que pueda haber en cuanto a los subdirectorios según se trabaje con Windows, MAC u otro sistema operativo. Finalmente, como la ubicación del programa WinBUGS puede ser distinta en cada ordenador, hemos guardado el objeto resultante (*ajuste.modelo*) para que posteriormente la persona puede cargarlo sin necesidad de correr el modelo él mismo, ganando así también carga computacional.

Finalmente, para el cuidado de la sintaxis nos hemos apoyado en la librería *formatR*, donde utilizando la función *tidy\_file*, corregimos posibles fallos sintácticos así como otorgamos del estilo adecuado a nuestro script.

```
tidy_file(file.path("../r", "PracticaDiseaseMapping.R"))
```