

Example 2.3

Disease mapping: from foundations to multidimensional modeling

Martinez-Beneito M.A. and Botella-Rocamora P.

This document reproduces the analysis made at Example 2.3 of the book: “Disease mapping: from foundations to multidimensional modeling” by Martinez-Beneito M.A. and Botella-Rocamora P., published by CRC press in 2019. You can watch the analysis made with full detail at this pdf document, or even execute it if you want with the material available at <https://github.com/MigueBeneito/DMBook>. Anyway, this pdf file should be enough for following most of the details of the analysis made for this example.

The statistical analysis below has been run in **R**, by additionally using the library **Rmarkdown**, so be sure that you have this software installed if you want to reproduce by yourself the content of this document. In that case we advise you to download first the annex material at <https://github.com/MigueBeneito/DMBook>, open with **Rstudio** the corresponding **.Rproj** file that you will find at the folder corresponding to this example and compile the corresponding **.Rmd** document. This will allow you to reproduce the whole statistical analysis below.

The inference carried out in this example has been undertaken with **WinBUGS**, that is introduced in Chapter 3 of the book. Nevertheless we report also the code for this example just in case you wanted to revisit it once you have read Chapter 3.

Libraries and data loading

```
# Libraries loading
#-----
if (!require(pbugs)) {
  if (!require(devtools)) {
    install.packages("devtools")
    devtools::install_github("fisabio/pbugs")
  } else {
    install_github("fisabio/pbugs")
  }
}

# Data loading
#-----
load("../Data/OralCancerTimeTrends.RData")
```

Data preparation

```
# covariate
year = 1991:2011
year.centered = year - mean(year)

# rates
rates = 1e+05 * O/Pop

# Gaussian bases of functions
base19 = matrix(nrow = 21, ncol = 19)
```

```

for (j in 1:19) {
  base19[, j] = dnorm(1:21, 1 + j, 1)
}
base9 = matrix(nrow = 21, ncol = 9)
for (j in 1:9) {
  base9[, j] = dnorm(1:21, 1 + j * 2, 2)
}

```

WinBUGS calls to fixed effects model

```

# WinBUGS model
model.fixed = function() {
  for (i in 1:n) {
    rate[i] ~ dnorm(mu[i], tau.rate)
    mu[i] <- beta[1] + beta[2] * year[i] + inprod2(base[i, ], gamma[])
  }
  for (i in 1:nBase) {
    gamma[i] ~ dflat()
  }
  beta[1] ~ dflat()
  beta[2] ~ dflat()

  tau.rate <- pow(sd.rate, -2)
  sd.rate ~ dunif(0, 10)
}

# WinBUGS run with a basis of 19 functions
data = list(rate = rates, year = year.centered, base = base19, nBase = 19,
  n = 21)
inits = function() {
  list(beta = rnorm(2), gamma = rnorm(19))
}
param = c("beta", "gamma", "sd.rate", "mu")
Resul19.fixed = pbugs(data = data, inits = inits, param = param, model = model.fixed,
  n.iter = 10000, n.burnin = 1000, n.thin = 9, DIC = FALSE, bugs.seed = 1)

# WinBUGS run with a basis of 9 functions
data = list(rate = rates, year = year.centered, base = base9, nBase = 9,
  n = 21)
inits = function() {
  list(beta = rnorm(2), gamma = rnorm(9))
}
param = c("beta", "gamma", "sd.rate", "mu")
Resul9.fixed = pbugs(data = data, inits = inits, param = param, model = model.fixed,
  n.iter = 10000, n.burnin = 1000, n.thin = 9, DIC = FALSE, bugs.seed = 1)

```

WinBUGS calls to random effects model

```

# WinBUGS model
model.random = function() {

```

```

    for (i in 1:n) {
      rate[i] ~ dnorm(mu[i], tau.rate)
      mu[i] <- beta[1] + beta[2] * year[i] + inprod2(base[i, ], gamma[])
    }
    for (i in 1:nBase) {
      gamma[i] ~ dnorm(0, tau.random)
    }
    beta[1] ~ dflat()
    beta[2] ~ dflat()

    tau.rate <- pow(sd.rate, -2)
    tau.random <- pow(sd.random, -2)
    sd.rate ~ dunif(0, 10)
    sd.random ~ dunif(0, 10)
  }

# WinBUGS run with a basis of 19 functions
data = list(rate = rates, year = year.centered, base = base19, nBase = 19,
  n = 21)
inits = function() {
  list(beta = rnorm(2), gamma = rnorm(19))
}
param = c("beta", "gamma", "sd.rate", "sd.random", "mu")
Resul19.random = pbugs(data = data, inits = inits, param = param, model = model.random,
  n.iter = 10000, n.burnin = 1000, n.thin = 9, DIC = FALSE, bugs.seed = 1)

# WinBUGS run with a basis of 9 functions
data = list(rate = rates, year = year.centered, base = base9, nBase = 9,
  n = 21)
inits = function() {
  list(beta = rnorm(2), gamma = rnorm(9))
}
param = c("beta", "gamma", "sd.rate", "sd.random", "mu")
Resul9.random = pbugs(data = data, inits = inits, param = param, model = model.random,
  n.iter = 10000, n.burnin = 1000, n.thin = 9, DIC = FALSE, bugs.seed = 1)

```

Plots

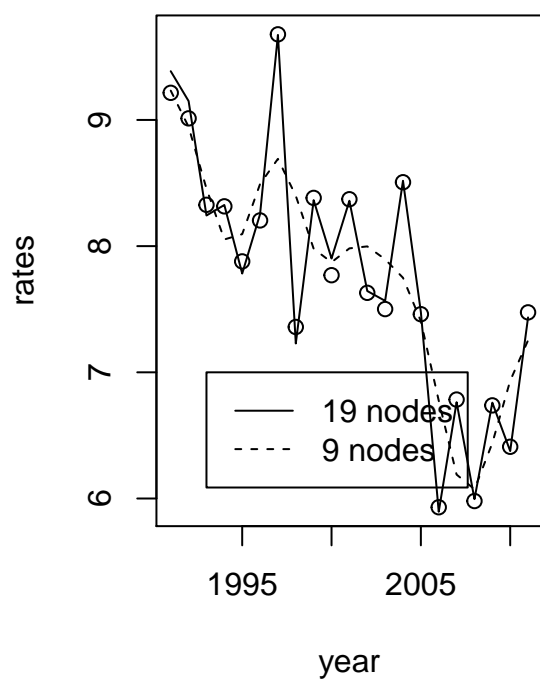
Plot the fit of both models for both two bases of functions

```

par(mfrow = c(1, 2))
# fixed effects fit
plot(year, rates)
lines(year, Resul19.fixed$mean$mu)
lines(year, Resul9.fixed$mean$mu, lty = 2)
legend(x = 1993, y = 7, lty = 1:2, legend = c("19 nodes", "9 nodes"))
title("Fixed effects model")
# random effects fit
plot(year, rates)
lines(year, Resul19.random$mean$mu)
lines(year, Resul9.random$mean$mu, lty = 2)
legend(x = 1993, y = 7, lty = 1:2, legend = c("19 nodes", "9 nodes"))
title("Random effects model")

```

Fixed effects model



Random effects model

