

# Example 2.4

Disease mapping: from foundations to multidimensional modeling

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

This document reproduces the analysis made at Example 2.4 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(cubature)) {
  install.packages("cubature")
  library(cubature)
}
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

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

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

## MCMC sampling with WinBUGS

```
# Bayesian linear regression model for WinBUGS
RegLin = function() {
  for (i in 1:n) {
    rate[i] ~ dnorm(media[i], prec)
    media[i] <- beta1 + beta2 * year[i]
  }
  prec <- pow(sigma, -2)
  sigma ~ dunif(0, 1000)
  beta1 ~ dnorm(0, 1e-06)
  beta2 ~ dnorm(0, 1e-06)
}

# WinBUGS call for making inference on the model above

# Data
data = list(year = year - mean(year), rate = rates, n = 21)

# Initial values
inits = function() {
  list(beta1 = rnorm(1, 0, 10), beta2 = rnorm(1, 0, 10), sigma = runif(1,
    0, 10))
}

# Parameters to save
parameters = c("beta1", "beta2", "sigma")

# WinBUGS call
RegLinWB = pbugs(data = data, inits = inits, param = parameters, model = RegLin,
  bugs.seed = 1, DIC = F, n.iter = 5000, n.burnin = 0, n.chains = 1,
  n.thin = 1)
```

### (Numeric) posterior distribution of sigma

Code adapted from Example 2.1

```
# Full posterior distribution without the unknown integration constant
post.full = function(y, x, beta1, beta2, sigma, sd.beta, sigma.up) {
  sigma^(-length(y)) * (sigma < sigma.up) * exp(-0.5 * ((beta1^2 + beta2^2)/(sd.beta^2) +
    sum((y - beta1 - beta2 * x)^2)/(sigma^2)))
}

# posterior distribution as a function of a single argument, as
# required for the numerical integration function adaptIntegrate
posterior = function(argum) {
  post.full(rates, year.centered, argum[1], argum[2], argum[3], 1000,
    1000)
}

# Integral of the full posterior distribution without integration
# constant as defined above.
total = hcubature(posterior, lowerLimit = c(6.8, -0.25, 0.3), upperLimit = c(8.7,
```

```

0, 1.8))$integral

# posterior distribution for sigma given the rest of parameters
post.sigma = function(sigma, other) {
  posterior(c(other[1], other[2], sigma))
}

# posterior densities for a grid of values for beta1, beta2 and sigma
range.sigma = 0.3 + (1:100) * 1.5/100
dsigma = vector()
for (i in 1:100) {
  dsigma[i] = adaptIntegrate(post.sigma, sigma = range.sigma[i], lowerLimit = c(6.8,
    -0.25), upperLimit = c(8.7, 0))$integral/total
}

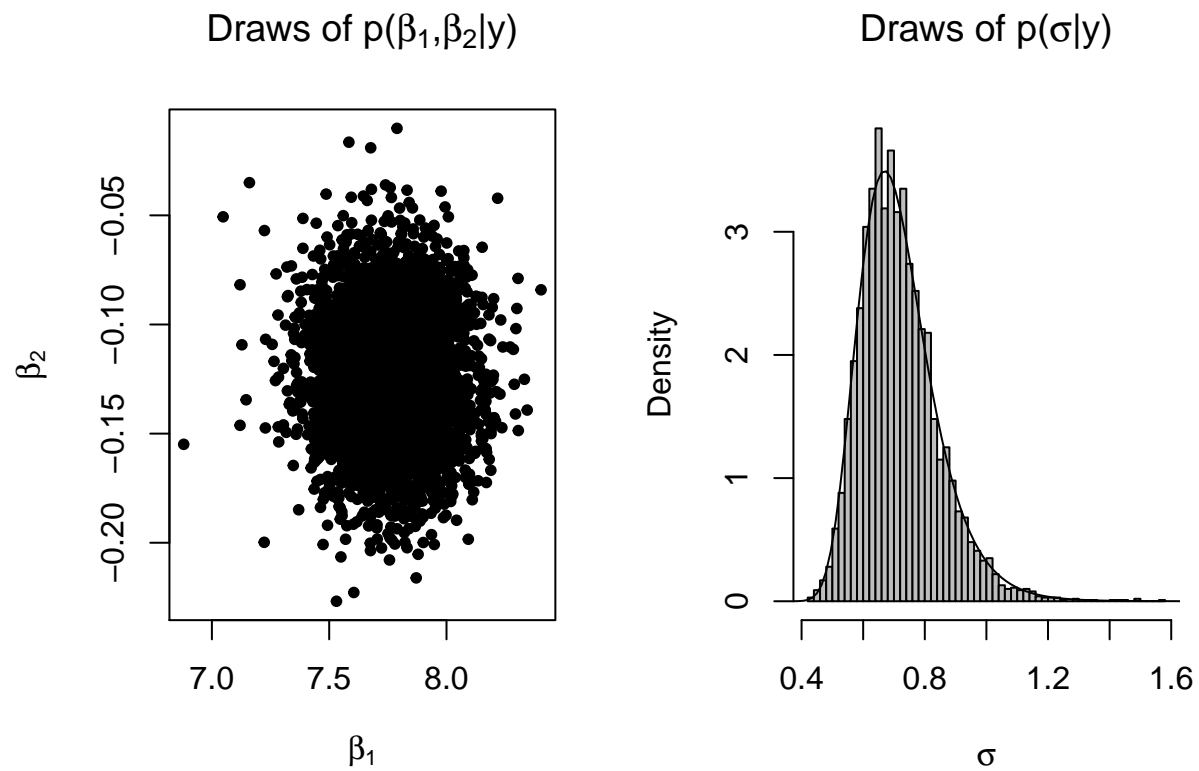
```

## Plotting of posterior samples

```

par(mfrow = c(1, 2))
# Posterior sample beta
plot(RegLinWB$sims.list$beta1, RegLinWB$sims.list$beta2, xlab = expression(beta[1]),
  ylab = expression(beta[2]), main = expression(paste("Draws of p(",
    beta[1], ",", beta[2], "|y)", sep = "")), pch = 20)
# Posterior sample sigma
hist(RegLinWB$sims.list$sigma, xlab = expression(sigma), prob = T, col = "gray",
  breaks = 70, main = expression(paste("Draws of p(", sigma, "|y)", sep = "")))
lines(range.sigma, dsigma)

```



## Posterior summaries

```
# Posterior summaries (including means, medians and 95% credible
# intervals)
round(RegLinWB$summary, 3)
```

```
##      mean    sd  2.5%   25%   50%   75%  97.5%
## beta1  7.760 0.161  7.452  7.655  7.756  7.868  8.078
## beta2 -0.124 0.027 -0.175 -0.141 -0.124 -0.106 -0.072
## sigma  0.718 0.126  0.522  0.630  0.700  0.786  1.010
```

```
# P(beta_2 < 0)
mean(RegLinWB$sims.list$beta2 < 0)
```

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