Example 2.5

Disease mapping: from foundations to multidimensional modeling

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

This document reproduces the analysis made at Example 2.5 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 will be introduced in the next chapter. The details for fully understanding this code are given at Chapter 3, nevertheless you may find interesting to take it a look at this code now, or later when you 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

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

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

MCMC sampling with WinBUGS (centered covariate)

```
RegLin2 = 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 ~ dt(0, 1e-06, 2)
    beta2 ~ dt(0, 1e-06, 2)
}

# WinBUGS call for making inference on the model above

# Data
data = list(year = year.centered, 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
RegLin2WB = pbugs(data = data, inits = inits, param = parameters, model = RegLin2,
    bugs.seed = 1, DIC = F, n.iter = 5500, n.burnin = 500, n.chains = 3,
    n.thin = 1)

# Posterior summaries
round(RegLin2WB$summary, 3)
```

```
##         mean    sd   2.5%    25%    50%    75%  97.5%  Rhat n.eff
## beta1  7.760 0.158  7.446  7.658  7.759  7.864  8.069 1.001 15000
## beta2 -0.124 0.026 -0.176 -0.141 -0.124 -0.107 -0.073 1.001 15000
## sigma  0.717 0.127  0.516  0.627  0.700  0.788  1.010 1.001  5300
```

MCMC sampling with WinBUGS (uncentered covariate)

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

# Initial values
inits = function() {
    list(beta1 = rnorm(1, 500, 20), beta2 = rnorm(1, -0.1, 0.1), sigma = runif(1,
        0, 4))
}

# WinBUGS call
```

```
RegLin2WB.uncentered = pbugs(data = data, inits = inits, param = parameters,
    model = RegLin2, bugs.seed = 1, DIC = F, n.iter = 5500, n.burnin = 500,
    n.chains = 3, n.thin = 1)

# Posterior summaries
round(RegLin2WB.uncentered$summary, 3)
```

```
##          mean      sd     2.5%      25%     50%     75%   97.5%   Rhat
## beta1 107.783 277.716 -251.700 -214.100 100.400 444.300 462.700 46.294
## beta2  -0.050   0.139   -0.227   -0.218  -0.046   0.111   0.130 46.301
## sigma   1.191   0.450    0.668    0.843   1.008   1.534   2.209  3.400
##       n.eff
## beta1     3
## beta2     3
## sigma     4
```

Long MCMC sampling with WinBUGS (uncentered covariate)

```
# WinBUGS call
RegLin2WB.uncentered2 = pbugs(data = data, inits = inits, param = parameters,
    model = RegLin2, bugs.seed = 1, DIC = F, n.iter = 2e+06, n.burnin = 2e+05,
    n.chains = 3, n.thin = 360)

# Posterior summaries
round(RegLin2WB.uncentered2$summary, 3)
```

```
##          mean     sd    2.5%     25%     50%     75%   97.5%  Rhat n.eff
## beta1 257.549 53.600 145.800 221.000 255.700 291.400 364.500 1.074    34
## beta2  -0.125  0.027  -0.178  -0.142  -0.124  -0.107  -0.069 1.068    34
## sigma   0.721  0.128   0.523   0.632   0.704   0.791   1.019 1.001 15000
```

```
cor(RegLin2WB.uncentered2$sims.list$beta1, RegLin2WB.uncentered2$sims.list$beta2)
```

```
## [1] -0.999995
```

Convergence plots

```
par(mfrow = c(2, 2))
plot(RegLin2WB$sims.array[, 1, 1], type = "l", main = expression(beta[1]),
    xlab = "Iteration", ylab = "Value")
lines(RegLin2WB$sims.array[, 2, 1], type = "l", col = gray(0.33))
lines(RegLin2WB$sims.array[, 3, 1], type = "l", col = gray(0.66))

plot(RegLin2WB.uncentered$sims.array[, 1, 1], type = "l", main = expression(beta[1]),
    xlab = "Iteration", ylab = "Value", ylim = c(min(RegLin2WB.uncentered$sims.array[,
        , 1]), max(RegLin2WB.uncentered$sims.array[, , 1])))
lines(RegLin2WB.uncentered$sims.array[, 2, 1], type = "l", col = gray(0.33))
lines(RegLin2WB.uncentered$sims.array[, 3, 1], type = "l", col = gray(0.66))

plot(RegLin2WB.uncentered2$sims.array[, 1, 1], type = "l", main = expression(beta[1]),
    xlab = "Iteration", ylab = "Value", ylim = c(min(RegLin2WB.uncentered2$sims.array[,
```

```
        , 1]), max(RegLin2WB.uncentered2$sims.array[, , 1])))
lines(RegLin2WB.uncentered2$sims.array[, 2, 1], type = "l", col = gray(0.33))
lines(RegLin2WB.uncentered2$sims.array[, 3, 1], type = "l", col = gray(0.66))

plot(RegLin2WB.uncentered2$sims.array[, 1, 2], type = "l", main = expression(beta[2]),
    xlab = "Iteration", ylab = "Value", ylim = c(min(RegLin2WB.uncentered2$sims.array[,
        , 2]), max(RegLin2WB.uncentered2$sims.array[, , 2])))
lines(RegLin2WB.uncentered2$sims.array[, 2, 2], type = "l", col = gray(0.33))
lines(RegLin2WB.uncentered2$sims.array[, 3, 2], type = "l", col = gray(0.66))
```