

Used code to obtain results

## 1. Execution of models in WinBUGS using the library R2WinBUGS

### Load libraries and data

```
# Working directory
DirMain = " " # Set an appropriate directory
setwd(DirMain)
# Load library and data
library(R2WinBUGS)
library(knitr)
library(pbugs) # For running the models in parallel calls to WinBUGS
load("datos.Rdata")
```

### BYM model

```
# BYM model with independent diseases,
# WinBUGS code
BYM.indep <- function() {
  for (j in 1:Ndiseases) {
    for (i in 1:Nareas) {
      O[i, j] ~ dpois(lambda[i, j])
      # Modeling of the mean for each
      # municipality
      log(lambda[i, j]) <- log(E[i,
        j]) + mu[j] + sdhet[j] *
        het[i, j] + sdsp[j] * sp[j,
        i]
      # SMR for each municipality
      SMR[i] <- exp(mu[j] + sdhet[j] *
        het[i, j] + sdsp[j] * sp[j,
        i])
      # Prior distribution for the non-spatial
      # effect
      het[i, j] ~ dnorm(0, 1)
    }
    # Prior distribution for the spatial
    # effect
    sp[j, 1:n] ~ car.normal(adj[], w[],
      num[], 1)
    # Prior distribution for the mean risk
    # for all municipalities
    mu[j] ~ dflat()
    # Prior distribution for the standard
    # deviations of the random effects
    sdhet[j] ~ dunif(0, 5)
```

```

    sdsp[j] ~ dunif(0, 5)
  }
}
# Run BYM model for each city, sex and
# disease
# City = 1: Alicante, 2: Castellón, 3:
# Valencia
for (i in 1:3) {
  # Specific mortality causes
  j <- 2
  # Sex = 1: Mens, 2: Women
  for (k in 1:2) {
    # Mortality cause
    l <- c(1:20)
    # Matrix with observed and expected cases
    O <- t(apply(Obs[[i]][[j]][k, , ,
      ], c(1, 2), sum)[1, ]))
    E <- t(apply(Esp[[i]][[j]][k, , ,
      ], c(1, 2), sum)[1, ]))
    Nareas <- dim(O)[1]
    Ndiseases <- dim(O)[2]
    # Data
    data <- list(Nareas = Nareas, Ndiseases = Ndiseases,
      O = O, E = E, adj = unlist(nb[[i]]),
      weights = rep(1, length(unlist(nb[[i]]))),
      num = sapply(nb[[i]], length))
    # Initial values
    initials <- function() {
      list(mu = rnorm(Ndiseases, 0,
        0.1), sdhet = runif(Ndiseases,
        0, 1), sdsp = runif(Ndiseases,
        0, 1), het = matrix(rnorm(Nareas *
        Ndiseases), nrow = Ndiseases,
        ncol = Nareas), sp = matrix(rnorm(Nareas *
        Ndiseases), nrow = Ndiseases,
        ncol = Nareas))
    }
    # Variables to retrieve
    param <- c("SMR", "lambda", "het",
      "sp", "mu", "sdsp", "sdhet")
    # Calls to WinBUGS
    t.result <- system.time(result <- pbugs(data = data,
      model.file = BYM.indep, inits = initials,
      parameters.to.save = param, n.chains = 3,
      n.iter = 30000, n.burnin = 5000,
      DIC = F))
    # Save results
    save(result, t.result, file = paste(getwd(),
      "/resul/resul.BYMIndep.", i,
      ".", j, ".", k, ".Rdata", sep = ""))
  }
}

```

## Fixed effects M-model (Section 2 in paper)

```
# Fixed effects M-model, WinBUGS code
Mmodel.FE <- function() {
  for (i in 1:Nareas) {
    for (j in 1:Ndiseases) {
      O[i, j] ~ dpois(lambda[i, j])
      # Modeling of the mean for each
      # municipality and disease
      log(lambda[i, j]) <- log(E[i,
        j]) + mu[j] + Theta[i, j]
      # SMR for each municipality and disease
      SMR[i, j] <- exp(mu[j] + Theta[i,
        j])
    }
  }
  # Definition of the random effects matrix
  for (i in 1:Nareas) {
    for (j in 1:Ndiseases) {
      Theta[i, j] <- inprod2(tPhi[,
        i], M[, j])
    }
  }
  # Matrix of spatially correlated random
  # effects if M is a square matrix define
  # Nsp (Number of spatial underlying
  # patterns) as Ndiseases
  for (j in 1:Nsp) {
    # Prior distribution for the spatial
    # effect
    Spatial[j, 1:Nareas] ~ car.normal(adj[,
      weights[], num[], 1)
    for (i in 1:Nareas) {
      # Prior distribution for the non-spatial
      # effect
      Het[j, i] ~ dnorm(0, 1)
      tPhi[j, i] <- Spatial[j, i]
    }
  }
  for (j in (Nsp + 1):(2 * Nsp)) {
    for (i in 1:Nareas) {
      tPhi[j, i] <- Het[(j - Nsp),
        i]
    }
  }
  # M-matrix
  for (i in 1:(2 * Nsp)) {
    for (j in 1:Ndiseases) {
      M[i, j] ~ dflat()
    }
  }
  # Others prior distributions
  for (j in 1:Ndiseases) {
    # Prior distribution for the mean risk
```

```

    # for all municipalities
    mu[j] ~ dflat()
  }
}
# Run fixed effects M-model considering
# 20 causes of mortality
# City = 1: Alicante, 2: Castellón, 3:
# Valencia
for (i in 1:3) {
  # Specific mortality causes
  j <- 2
  # Sex = 1: Mens, 2: Women
  for (k in 1:2) {
    # Mortality causes
    l <- c(1:20)
    # Matrix with observed and expected cases
    O <- t(apply(Obs[[i]][[j]][k, , ,
      ], c(1, 2), sum)[1, ]
    E <- t(apply(Esp[[i]][[j]][k, , ,
      ], c(1, 2), sum)[1, ]
    Nareas <- dim(O)[1]
    Ndiseases <- dim(O)[2]
    # Data
    data <- list(Nareas = Nareas, Ndiseases = Ndiseases,
      Nsp = Ndiseases, O = O, E = E,
      adj = unlist(nb[[i]]), weights = rep(1,
        length(unlist(nb[[i]]))),
      num = sapply(nb[[i]], length))
    # Initial values
    initials <- function() {
      list(mu = rnorm(Ndiseases, 0,
        0.1), M = matrix(rnorm(2 *
          Nsp * Ndiseases), nrow = 2 *
          Nsp, ncol = Ndiseases), Het = matrix(rnorm(Nareas *
            Nsp), nrow = Nsp, ncol = Nareas),
          Spatial = matrix(rnorm(Nareas *
            Nsp), nrow = Nsp, ncol = Nareas))
    }
    # Variables to retrieve
    param <- c("SMR", "lambda", "M",
      "Het", "Spatial", "mu", "Theta")
    # Calls to WinBUGS
    t.result <- system.time(result <- pbugs(data = data,
      model.file = Mmodel.FE, inits = initials,
      parameters.to.save = param, n.chains = 3,
      n.iter = 30000, n.burnin = 5000,
      DIC = F))
    # Save results
    save(result, t.result, file = paste(getwd(),
      "/resul/resul.MmodelFE.", i,
      ".", j, ".", k, ".Rdata", sep = ""))
  }
}

```

## Random effects M-model or NVA M-model (Section 2 in paper)

```
# Random effects M-model, WinBUGS code
Mmodel.RE <- function() {
  for (i in 1:Nareas) {
    for (j in 1:Ndiseases) {
      O[i, j] ~ dpois(lambda[i, j])
      # Modeling of the mean for each
      # municipality and disease
      log(lambda[i, j]) <- log(E[i,
        j]) + mu[j] + Theta[i, j]
      # SMR for each municipality and disease
      SMR[i, j] <- exp(mu[j] + Theta[i,
        j])
    }
  }
  # Definition of the random effects matrix
  for (i in 1:Nareas) {
    for (j in 1:Ndiseases) {
      Theta[i, j] <- inprod2(tPhi[,
        i], M[, j])
    }
  }
  # Matrix of spatially correlated random
  # effects: if M is a square matrix define
  # Nsp (Number of spatial underlying
  # patterns) as Ndiseases
  for (j in 1:Nsp) {
    # Prior distribution for the spatial
    # effect
    Spatial[j, 1:Nareas] ~ car.normal(adj[,
      weights[], num[], 1)
    for (i in 1:Nareas) {
      # Prior distribution for the non-spatial
      # effect
      Het[j, i] ~ dnorm(0, 1)
      tPhi[j, i] <- Spatial[j, i]
    }
  }
  for (j in (Nsp + 1):(2 * Nsp)) {
    for (i in 1:Nareas) {
      tPhi[j, i] <- Het[(j - Nsp),
        i]
    }
  }
  # M-matrix
  for (j in 1:Ndiseases) {
    for (i in 1:Nsp) {
      M[i, j] ~ dnorm(0, prec.sp)
    }
    for (i in (Nsp + 1):(2 * Nsp)) {
      M[i, j] ~ dnorm(0, prec.het)
    }
  }
}
```

```

# Others prior distributions
# Prior distribution for the mean risk
# for all municipalities
for (j in 1:Ndiseases) {
  mu[j] ~ dflat()
}
# Prior distribution for the standard
# deviations of the random effects
prec.sp <- pow(sd.sp, -2)
sd.sp ~ dunif(0, 100)
prec.het <- pow(sd.het, -2)
sd.het ~ dunif(0, 100)
}

# Run random effects M-model considering
# 20 causes of mortality
# City = 1: Alicante, 2: Castellón, 3:
# Valencia
for (i in 1:3) {
  # Specific mortality causes
  j <- 2
  # Sex = 1: Mens, 2: Women
  for (k in 1:2) {
    # Mortality causes
    l <- c(1:20)
    # Matrix with observed and expected cases
    O <- t(apply(Obs[[i]][[j]][k, , ,
      ], c(1, 2), sum)[1, ]
    E <- t(apply(Esp[[i]][[j]][k, , ,
      ], c(1, 2), sum)[1, ]
    Nareas <- dim(O)[1]
    Ndiseases <- dim(O)[2]
    # Data
    data <- list(Nareas = Nareas, Ndiseases = Ndiseases,
      Nsp = Ndiseases, O = O, E = E,
      adj = unlist(nb[[i]]), weights = rep(1,
        length(unlist(nb[[i]]))),
      num = sapply(nb[[i]], length))
    # Initial values
    initials <- function() {
      list(mu = rnorm(Ndiseases, 0,
        0.1), sd.sp = runif(1, 0,
        1), sd.het = runif(1, 0,
        1), Het = matrix(rnorm(Nareas *
        Nsp), nrow = Nsp, ncol = Nareas),
        Spatial = matrix(rnorm(Nareas *
        Nsp), nrow = Nsp, ncol = Nareas))
    }
    # Variables to retrieve
    param <- c("sd.sp", "sd.het", "SMR",
      "lambda", "M", "Het", "Spatial",
      "mu")
    # Calls to WinBUGS
    t.result <- system.time(result <- pbugs(data = data,

```

```

        model.file = Mmodel.RE, inits = initials,
        parameters.to.save = param, n.chains = 3,
        n.iter = 30000, n.burnin = 5000,
        DIC = F))
    # Save results
    save(result, t.result, file = paste(getwd(),
        "/resul/resul.Mmodel.RE.", i,
        ".", j, ".", k, ".Rdata", sep = ""))
  }
}

```

## RVA M-model (Section 4 in paper)

```

# RVA M-model, WinBUGS code
Mmodel.RVA <- function() {
  for (i in 1:Nareas) {
    for (j in 1:Ndiseases) {
      O[i, j] ~ dpois(lambda[i, j])
      # Modeling of the mean for each
      # municipality and disease
      log(lambda[i, j]) <- log(E[i,
        j]) + mu[j] + Theta[i, j]
      # SMR for each municipality and disease
      SMR[i, j] <- exp(mu[j] + Theta[i,
        j])
    }
  }
  # Definition of the random effects matrix
  for (i in 1:Nareas) {
    for (j in 1:Ndiseases) {
      Theta[i, j] <- inprod2(tPhi[,
        i], M[, j])
    }
  }
  # Matrix of spatially correlated random
  # effects if M is a square matrix define
  # Nsp (Number of spatial underlying
  # patterns) as Ndiseases
  for (j in 1:Nsp) {
    # Prior distribution for the spatial
    # effect
    Spatial[j, 1:Nareas] ~ car.normal(adj[,
      weights[], num[], 1)
    for (i in 1:Nareas) {
      # Prior distribution for the non-spatial
      # effect
      Het[j, i] ~ dnorm(0, 1)
      tPhi[j, i] <- Spatial[j, i]
    }
  }
  for (j in (Nsp + 1):(2 * Nsp)) {
    for (i in 1:Nareas) {
      tPhi[j, i] <- Het[(j - Nsp),

```

```

        i]
    }
}
# M-matrix
for (j in 1:Ndiseases) {
  for (i in 1:(2 * Nsp)) {
    M.aux[i, j] ~ dnorm(0, 1)
    M[i, j] <- sd[i] * M.aux[i, j]
  }
}
# Others prior distributions
# Prior distribution for the mean risk
# for all municipalities
for (j in 1:Ndiseases) {
  mu[j] ~ dflat()
}
# Prior distribution for the standard
# deviations of the random effects
for (i in 1:(2 * Nsp)) {
  sd[i] ~ dunif(0, 5)
}
}
# Run RVA M-model considering 20 causes
# of mortality
# City = 1: Alicante, 2: Castellón, 3:
# Valencia
for (i in 1:3) {
  # Specific mortality causes
  j <- 2
  # Sex = 1: Mens, 2: Women
  for (k in 1:2) {
    # Mortality causes
    l <- c(1:20)
    # Matrix with observed and expected cases
    O <- t(apply(Obs[[i]][[j]][k, , ,
      ], c(1, 2), sum)[l, ])
    E <- t(apply(Esp[[i]][[j]][k, , ,
      ], c(1, 2), sum)[l, ])
    Nareas <- dim(O)[1]
    Ndiseases <- dim(O)[2]
    # Data
    data <- list(Nareas = Nareas, Ndiseases = Ndiseases,
      Nsp = Ndiseases, O = O, E = E,
      adj = unlist(nb[[i]]), weights = rep(1,
        length(unlist(nb[[i]]))),
      num = sapply(nb[[i]], length))
    # Initial values
    initials <- function() {
      list(mu = rnorm(Ndiseases, 0,
        0.1), sd = runif(2 * Nsp,
        0.1, 1), Het = matrix(rnorm(Nareas *
        Nsp), nrow = Nsp, ncol = Nareas),
        Spatial = matrix(rnorm(Nareas *

```



```

        Nsp), nrow = Nsp, ncol = Nareas))
    }
    # Variables to retrieve
    param <- c("sd", "SMR", "lambda",
              "M", "mu")
    # Calls to WinBUGS
    t.result <- system.time(result <- pbugs(data = data,
      model.file = Mmodel.RVA, inits = initials,
      parameters.to.save = param, n.chains = 3,
      n.iter = 30000, n.burnin = 5000,
      DIC = F))
    # Save results
    save(result, t.result, file = paste(getwd(),
      "/resul/resul.MmodelRVA.", i,
      ".", j, ".", k, ".Rdata", sep = ""))
  }
}

```

## CVA M-model (Section 4 in paper)

```

# CVA M-model, WinBUGS code
Mmodel.CVA <- function() {
  for (i in 1:Nareas) {
    for (j in 1:Ndiseases) {
      O[i, j] ~ dpois(lambda[i, j])
      # Modeling of the mean for each
      # municipality and disease
      log(lambda[i, j]) <- log(E[i,
        j]) + mu[j] + Theta[i, j]
      # SMR for each municipality and disease
      SMR[i, j] <- exp(mu[j] + Theta[i,
        j])
    }
  }
  # Definition of the random effects matrix
  for (i in 1:Nareas) {
    for (j in 1:Ndiseases) {
      Theta[i, j] <- inprod2(tPhi[,
        i], M[, j])
    }
  }
  # Matrix of spatially correlated random
  # effects if M is a square matrix define
  # Nsp (Number of spatial underlying
  # patterns) as Ndiseases
  for (j in 1:Nsp) {
    # Prior distribution for the spatial
    # effect
    Spatial[j, 1:Nareas] ~ car.normal(adj[,
      weights[], num[], 1)
    for (i in 1:Nareas) {
      # Prior distribution for the non-spatial
      # effect

```

```

        Het[j, i] ~ dnorm(0, 1)
        tPhi[j, i] <- Spatial[j, i]
    }
}
for (j in (Nsp + 1):(2 * Nsp)) {
    for (i in 1:Nareas) {
        tPhi[j, i] <- Het[(j - Nsp),
            i]
    }
}
# M-matrix
for (j in 1:Ndiseases) {
    for (i in 1:Nsp) {
        M.aux[i, j] ~ dnorm(0, 1)
        M[i, j] <- sdstruct.sp[j] * M.aux[i,
            j]
    }
    for (i in (Nsp + 1):(2 * Nsp)) {
        M.aux[i, j] ~ dnorm(0, 1)
        M[i, j] <- sdstruct.het[j] *
            M.aux[i, j]
    }
}
# Others prior distributions Prior
# distribution for the mean risk for all
# municipalities
for (j in 1:Ndiseases) {
    mu[j] ~ dflat()
}
# Prior distribution for the standard
# deviations of the random effects
for (j in 1:Ndiseases) {
    prec.sp[j] <- pow(sdstruct.sp[j],
        -2)
    sdstruct.sp[j] ~ dunif(0, 5)
    prec.het[j] <- pow(sdstruct.het[j],
        -2)
    sdstruct.het[j] ~ dunif(0, 5)
}
}
# Run CVA M-model considering 20 causes
# of mortality
# City = 1: Alicante, 2: Castellón, 3:
# Valencia
for (i in 1:3) {
    # Specific mortality causes
    j <- 2
    # Sex = 1: Mens, 2: Women
    for (k in 1:2) {
        # Mortality causes
        l <- c(1:20)
        # Matrix with observed and expected cases
        O <- t(apply(Obs[[i]][[j]][k, , ,

```

```

    ], c(1, 2), sum)[1, ])
E <- t(apply(Esp[[i]][[j]][k, , ,
    ], c(1, 2), sum)[1, ])
Nareas <- dim(O)[1]
Ndiseases <- dim(O)[2]
# Data
data <- list(Nareas = Nareas, Ndiseases = Ndiseases,
  Nsp = Ndiseases, O = O, E = E,
  adj = unlist(nb[[i]]), weights = rep(1,
    length(unlist(nb[[i]]))),
  num = sapply(nb[[i]], length))
# Initial values
initials <- function() {
  list(mu = rnorm(Ndiseases, 0,
    0.1), sdstruct.sp = runif(Ndiseases,
    0, 1), sdstruct.het = runif(Ndiseases,
    0, 1), Het = matrix(rnorm(Nareas *
    Nsp), nrow = Nsp, ncol = Nareas),
    Spatial = matrix(rnorm(Nareas *
    Nsp), nrow = Nsp, ncol = Nareas))
}
# Variables to retrieve
param <- c("sdstruct.sp", "sdstruct.het",
  "SMR", "lambda", "M", "mu")
# Calls to WinBUGS
t.result <- system.time(result <- pbugs(data = data,
  model.file = Mmodel.CVA, inits = initials,
  parameters.to.save = param, n.chains = 3,
  n.iter = 30000, n.burnin = 5000,
  DIC = F))
# Save results
save(result, t.result, file = paste(getwd(),
  "/resul/resul.MmodelCVA.", i,
  ".", j, ".", k, ".Rdata", sep = ""))
}
}

```

## 2. Choropleth maps for all models

```

# Load libraries
library(RColorBrewer)
cuts_SMR <- c(0, 0.67, 0.8, 0.91, 1.1, 1.25,
  1.5)
palette <- brewer.pal(7, "BrBG")[7:1]
# Name of mortality causes
Causes <- dimnames(Obs[[i]][[j]])[[2]]
# City = 1: Alicante, 2: Castellón, 3:
# Valencia
for (i in 1:3) {
  # Specific mortality causes
  j <- 2
  # Sex = 1: Mens, 2: Women

```

```

k <- 1
# Mortality causes
for (l in 1:20) {
  # Load WinBUGS results, BYM model
  load(paste(getwd(), "/resul.BYMIndep.",
    i, ".", j, ".", k, ".", l, ".Rdata",
    sep = ""))
  # SMR estimates, BYM model
  aux <- palette[findInterval(result$mean$SMR,
    cuts_SMR)]
  plot(Carto[[i]], col = palette[aux],
    main = paste0("BYM - ", Causas[l]),
    lwd = 0.2)
  legend("bottomright", c("< 0.67",
    "0.67 - 0.80", "0.80 - 0.91",
    "0.91 - 1.10", "1.10 - 1.25",
    "1.25 - 1.50", "> 1.50"), title = "SMR",
    border = NULL, fill = palette,
    bty = "n")
  # Load WinBUGS results, fixed effects
  # M-model
  load(paste(getwd(), "/resul.MmodelFE.",
    i, ".", j, ".", k, ".Rdata",
    sep = ""))
  # SMR estimates, fixed effects M-model
  aux <- palette[findInterval(result$mean$SMR[,
    l], cuts_SMR)]
  plot(Carto[[i]], col = palette[aux],
    main = paste0("MmodelFE - ",
    Causas[l]), lwd = 0.2)
  legend("bottomright", c("< 0.67",
    "0.67 - 0.80", "0.80 - 0.91",
    "0.91 - 1.10", "1.10 - 1.25",
    "1.25 - 1.50", "> 1.50"), title = "SMR",
    border = NULL, fill = palette,
    bty = "n")
  # Load WinBUGS results, random effects
  # M-model
  load(paste(getwd(), "/resul.MmodelRE.",
    i, ".", j, ".", k, ".Rdata",
    sep = ""))
  # SMR estimates, random effects M-model
  aux <- palette[findInterval(result$mean$SMR[,
    l], cuts_SMR)]
  plot(Carto[[i]], col = palette[aux],
    main = paste0("MmodelRE - ",
    Causas[l]), lwd = 0.2)
  legend("bottomright", c("< 0.67",
    "0.67 - 0.80", "0.80 - 0.91",
    "0.91 - 1.10", "1.10 - 1.25",
    "1.25 - 1.50", "> 1.50"), title = "SMR",
    border = NULL, fill = palette,
    bty = "n")
}

```

```

# Load WinBUGS results, RVA m-model
load(paste(getwd(), "/resul.MmodelRVA.",
  i, ".", j, ".", k, ".Rdata",
  sep = ""))
# SMR estimates, RVA M-model
aux <- palette[findInterval(result$mean$SMR[,
  1], cuts_SMR)]
plot(Carto[[i]], col = palette[aux],
  main = paste0("MmodelRVA - ",
    Causas[1]), lwd = 0.2)
legend("bottomright", c("< 0.67",
  "0.67 - 0.80", "0.80 - 0.91",
  "0.91 - 1.10", "1.10 - 1.25",
  "1.25 - 1.50", "> 1.50"), title = "SMR",
  border = NULL, fill = palette,
  bty = "n")
# Load WinBUGS results, CVA m-model
load(paste(getwd(), "/resul.MmodelCVA.",
  i, ".", j, ".", k, ".Rdata",
  sep = ""))
# SMR estimates, CVA M-model
aux <- palette[findInterval(result$mean$SMR[,
  1], cuts_SMR)]
plot(Carto[[i]], col = palette[aux],
  main = paste0("MmodelCVA - ",
    Causas[1]), lwd = 0.2)
legend("bottomright", c("< 0.67",
  "0.67 - 0.80", "0.80 - 0.91",
  "0.91 - 1.10", "1.10 - 1.25",
  "1.25 - 1.50", "> 1.50"), title = "SMR",
  border = NULL, fill = palette,
  bty = "n")
}
}

```

### 3. DIC for each model (Section 5, Table 1 in paper)

```

# Function for DICs calculation
CalculaDIC <- function(Simu, 0, save = FALSE) {
  mu <- Simu$sims.matrix[, which(substr(dimnames(Simu$sims.matrix)[[2]],
    1, 2) == "la")]
  D <- apply(mu, 1, function(x) {
    -2 * sum(dpois(as.vector(t(0)), x,
      log = T))
  })
  Dmedia <- mean(D)
  mumedia <- apply(mu, 2, mean)
  DenMedia <- -2 * sum(dpois(as.vector(t(0)),
    mumedia, log = T))
  if (save == TRUE) {
    return(c(Dmedia, Dmedia - DenMedia,
      2 * Dmedia - DenMedia))
  }
}

```

```

}
cat("D=", Dmedia, "pD=", Dmedia - DenMedia,
    "DIC=", 2 * Dmedia - DenMedia, "\n")
}
DIC.BYMIndep <- list()
DIC.MmodelFE <- list()
DIC.MmodelRE <- list()
DIC.MmodelRVA <- list()
DIC.MmodelCVA <- list()
# City = 1: Alicante, 2: Castellón, 3:
# Valencia
for (i in 1:3) {
  # Specific mortality causes
  j <- 2
  # Sex = 1: Mens, 2: Women
  k <- 1
  # Mortality causes
  l <- c(1:20)
  # Matrix with observed cases
  O <- t(apply(Obs[[i]][[j]][k, , ],
              c(1, 2), sum)[l, ])
  # DIC M-model with independent diseases
  load(paste(getwd(), "/resul.BYMIndep.",
             i, ".", j, ".", k, ".Rdata", sep = ""))
  DIC.BYMIndep[[i]] <- CalculaDIC(Simu = result,
                                O = O, save = TRUE)[3]
  # DIC fixed effects M-model
  load(paste(getwd(), "/resul.MmodelFE.",
             i, ".", j, ".", k, ".Rdata", sep = ""))
  DIC.MmodelFE[[i]] <- CalculaDIC(Simu = result,
                                O = O, save = TRUE)[3]
  # DIC random effects M-model
  load(paste(getwd(), "/resul.MmodelRE.",
             i, ".", j, ".", k, ".Rdata", sep = ""))
  DIC.MmodelRE[[i]] <- CalculaDIC(Simu = result,
                                O = O, save = TRUE)[3]
  # DIC RVA M-model
  load(paste(getwd(), "/resul.MmodelRVA.",
             i, ".", j, ".", k, ".Rdata", sep = ""))
  DIC.MmodelRVA[[i]] <- CalculaDIC(Simu = result,
                                O = O, save = TRUE)[3]
  # DIC CVA M-model
  load(paste(getwd(), "/resul.MmodelCVA.",
             i, ".", j, ".", k, ".Rdata", sep = ""))
  DIC.MmodelCVA[[i]] <- CalculaDIC(Simu = result,
                                O = O, save = TRUE)[3]
}
Table <- matrix(c(unlist(DIC.BYMIndep), unlist(DIC.MmodelFE),
                  unlist(DIC.MmodelRE), unlist(DIC.MmodelRVA),
                  unlist(DIC.MmodelCVA)), ncol = 3, byrow = TRUE)
rownames(Table) <- c("BYM with independent diseases",
                    "Fixed effects $M$-model", "Random effects $M$-model",
                    "RVA $M$-model", "CVA $M$-model")

```

```
colnames(Table) <- c("Alicante", "Castellón",
  "Valencia")
print(kable(Table, caption = "DICs for the adjusted models in each study city"))
```

#### 4. Log-likelihood of the predictive distribution of each model in the cases observed in odd years (Section 5, Table 2 in paper)

```
log_predict.BYMIndep <- array(dim = c(3,
  length(1), result$n.sims))
log_predict.MmodelFE <- array(dim = c(3,
  length(1), result$n.sims))
log_predict.MmodelRE <- array(dim = c(3,
  length(1), result$n.sims))
log_predict.MmodelRVA <- array(dim = c(3,
  length(1), result$n.sims))
log_predict.MmodelCVA <- array(dim = c(3,
  length(1), result$n.sims))
# City = 1: Alicante, 2: Castellón, 3:
# Valencia
for (i in 1:3) {
  # Specific mortality causes
  j <- 2
  # Sex = 1: Mens, 2: Women
  k <- 1
  # Mortality causes
  l <- c(1:20)
  # Matrix with observed and expected cases
  # in odd years
  O.odd <- t(apply(Obs[[i]][[j]][k, , ,
    seq(2, 12, by = 2)], c(1, 2), sum)[1,
    ])
  E.odd <- t(apply(Esp[[i]][[j]][k, , ,
    seq(2, 12, by = 2)], c(1, 2), sum)[1,
    ])
  # Load results M-model with independent
  # diseases (couple years)
  load(paste(getwd(), "/resul.BYMIndep.couple.",
    i, ".", j, ".", k, ".Rdata", sep = ""))
  # Load results fixed effects M-model
  # (couple years)
  load(paste(getwd(), "/resul.MmodelFE.couple.",
    i, ".", j, ".", k, ".Rdata", sep = ""))
  # Load results random effects M-model
  # (couple years)
  load(paste(getwd(), "/resul.MmodelRE.couple.",
    i, ".", j, ".", k, ".Rdata", sep = ""))
  # Load results RVA M-model (couple years)
  load(paste(getwd(), "/resul.MmodelRVA.couple.",
    i, ".", j, ".", k, ".Rdata", sep = ""))
  # Load results CVA M-model (couple years)
  load(paste(getwd(), "/resul.MmodelCVA.couple.",
    i, ".", j, ".", k, ".Rdata", sep = ""))
```

```

for (m in 1:length(l)) {
  for (n in 1:result$n.sims) {
    # M-model with independent diseases
    lambda <- E.odd[, m] * resul.BYMIndep.couple$sims.list$SMR[n,
      , m]
    log_predict.BYMIndep[i, m, n] <- sum(dpois(0.odd[,
      m], lambda, log = TRUE))
    # Fixed effects M-model
    lambda <- E.odd[, m] * resul.MmodelFE.couple$sims.list$SMR[n,
      , m]
    log_predict.MmodelFE[i, m, n] <- sum(dpois(0.odd[,
      m], lambda, log = TRUE))
    # Random effects M-model
    lambda <- E.odd[, m] * resul.MmodelRE.couple$sims.list$SMR[n,
      , m]
    log_predict.MmodelRE[i, m, n] <- sum(dpois(0.odd[,
      m], lambda, log = TRUE))
    # RVA M-model
    lambda <- E.odd[, m] * resul.MmodelRVA.couple$sims.list$SMR[n,
      , m]
    log_predict.MmodelRVA[i, m, n] <- sum(dpois(0.odd[,
      m], lambda, log = TRUE))
    # CVA M-model
    lambda <- E.odd[, m] * resul.MmodelCVA.couple$sims.list$SMR[n,
      , m]
    log_predict.MmodelCVA[i, m, n] <- sum(dpois(0.odd[,
      m], lambda, log = TRUE))
  }
}
}
Table <- matrix(c(apply(log_predict.BYMIndep,
  1, mean), apply(log_predict.MmodelFE,
  1, mean), apply(log_predict.MmodelRE,
  1, mean), apply(log_predict.MmodelRVA,
  1, mean), apply(log_predict.MmodelCVA,
  1, mean)), ncol = 3, byrow = TRUE)
rownames(Table) <- c("BYM with independent diseases",
  "Fixed effects $M$-model", "Random effects $M$-model",
  "RVA $M$-model", "CVA $M$-model")
colnames(Table) <- c("Alicante", "Castellón",
  "Valencia")
print(kable(Table, caption = "Log-likelihood of the predictive distribution of each model
  in the cases observed in odd years and in each study city"))

```