

Waller SISMID 2021: NY Leukemia Spatial Regression with INLA

Lance A. Waller

7/11/2021

New York Leukemia Data Spatial Poisson Regression with INLA Example

NY Leukemia analysis with INLA from https://www.rdocumentation.org/packages/spdep/versions/0.6-15/topics/NY_data and <https://www.r-bloggers.com/spatial-data-analysis-with-inla/>

```
##Load libraries
```

```
library(spdep)
```

```
## Loading required package: sp
```

```
## Loading required package: spData
```

```
## To access larger datasets in this package, install the spDataLarge
```

```
## package with: `install.packages('spDataLarge',
```

```
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
## Loading required package: sf
```

```
## Linking to GEOS 3.7.2, GDAL 2.4.2, PROJ 5.2.0
```

```
library(INLA)
```

```
## Loading required package: Matrix
```

```
## Loading required package: parallel
```

```
## Loading required package: foreach
```

```
## This is INLA_20.03.17 built 2020-07-09 19:30:17 UTC.
```

```
## See www.r-inla.org/contact-us for how to get help.
```

```
## To enable PARDISO sparse library; see inla.pardiso()
```

```
# Set my working directory (Lance's here for example)
```

```
setwd("~/OneDrive - Emory University/meetings/SISMID.2021/SISMID.2021.Waller.Rcode")
```

```
# Read in data (included in the 'foreign' library.)
```

```
library(foreign)
```

```
nydata <- read.dbf(system.file("etc/misc/nydata.dbf", package="spdep")[1])
```

Now format the data for INLA (this needs to be more clearly documented, but these steps put the right values from the dataset into the variable names needed for INLA)

```
# Format data for INLA
```

```
coordinates(nydata) <- c("X", "Y")
```

```
nyadjmat <- as.matrix(read.dbf(system.file("etc/misc/nyadjwts.dbf",  
                                           package="spdep")[1])[-1])
```

```
## Field name: 'Z600701190' changed to: 'Z600701190.1'
```

```

## Field name: 'Z600701190' changed to: 'Z600701190.2'
## Field name: 'Z600701200' changed to: 'Z600701200.1'
## Field name: 'Z600701210' changed to: 'Z600701210.1'
## Field name: 'Z600701210' changed to: 'Z600701210.2'
## Field name: 'Z600701220' changed to: 'Z600701220.1'
## Field name: 'Z600701270' changed to: 'Z600701270.1'
## Field name: 'Z600701320' changed to: 'Z600701320.1'
## Field name: 'Z600701330' changed to: 'Z600701330.1'
## Field name: 'Z605303010' changed to: 'Z605303010.1'
## Field name: 'Z605303010' changed to: 'Z605303010.2'
## Field name: 'Z605303040' changed to: 'Z605303040.1'
## Field name: 'Z605303040' changed to: 'Z605303040.2'
## Field name: 'Z605303050' changed to: 'Z605303050.1'
## Field name: 'Z606700170' changed to: 'Z606700170.1'
## Field name: 'Z606700360' changed to: 'Z606700360.1'
## Field name: 'Z606700560' changed to: 'Z606700560.1'
## Field name: 'Z606700610' changed to: 'Z606700610.1'
## Field name: 'Z606700610' changed to: 'Z606700610.2'
## Field name: 'Z606701100' changed to: 'Z606701100.1'
## Field name: 'Z606701110' changed to: 'Z606701110.1'
## Field name: 'Z606701120' changed to: 'Z606701120.1'
## Field name: 'Z606701120' changed to: 'Z606701120.2'
## Field name: 'Z606701120' changed to: 'Z606701120.3'
## Field name: 'Z606701140' changed to: 'Z606701140.1'
## Field name: 'Z606701520' changed to: 'Z606701520.1'
## Field name: 'Z606701650' changed to: 'Z606701650.1'
## Field name: 'Z606701680' changed to: 'Z606701680.1'
## Field name: 'Z606701690' changed to: 'Z606701690.1'

```

```

ID <- as.character(names(read.dbf(system.file("etc/misc/nyadjwts.dbf",
                                             package="spdep")[1]))[-1])

```

```

## Field name: 'Z600701190' changed to: 'Z600701190.1'
## Field name: 'Z600701190' changed to: 'Z600701190.2'
## Field name: 'Z600701200' changed to: 'Z600701200.1'
## Field name: 'Z600701210' changed to: 'Z600701210.1'
## Field name: 'Z600701210' changed to: 'Z600701210.2'
## Field name: 'Z600701220' changed to: 'Z600701220.1'

```

```
## Field name: 'Z600701270' changed to: 'Z600701270.1'
## Field name: 'Z600701320' changed to: 'Z600701320.1'
## Field name: 'Z600701330' changed to: 'Z600701330.1'
## Field name: 'Z605303010' changed to: 'Z605303010.1'
## Field name: 'Z605303010' changed to: 'Z605303010.2'
## Field name: 'Z605303040' changed to: 'Z605303040.1'
## Field name: 'Z605303040' changed to: 'Z605303040.2'
## Field name: 'Z605303050' changed to: 'Z605303050.1'
## Field name: 'Z606700170' changed to: 'Z606700170.1'
## Field name: 'Z606700360' changed to: 'Z606700360.1'
## Field name: 'Z606700560' changed to: 'Z606700560.1'
## Field name: 'Z606700610' changed to: 'Z606700610.1'
## Field name: 'Z606700610' changed to: 'Z606700610.2'
## Field name: 'Z606701100' changed to: 'Z606701100.1'
## Field name: 'Z606701110' changed to: 'Z606701110.1'
## Field name: 'Z606701120' changed to: 'Z606701120.1'
## Field name: 'Z606701120' changed to: 'Z606701120.2'
## Field name: 'Z606701120' changed to: 'Z606701120.3'
## Field name: 'Z606701140' changed to: 'Z606701140.1'
## Field name: 'Z606701520' changed to: 'Z606701520.1'
## Field name: 'Z606701650' changed to: 'Z606701650.1'
## Field name: 'Z606701680' changed to: 'Z606701680.1'
## Field name: 'Z606701690' changed to: 'Z606701690.1'
```

```
identical(substring(ID, 2, 10), substring(as.character(nydata$AREakey), 2, 10))
```

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

```
nyadjlw <- mat2listw(nyadjmat, as.character(nydata$AREakey))
listw_NY <- nb2listw(nyadjlw$neighbours, style="B")
```

Now let's calculate the overall rate (total number of cases over total population size).

Next, find the number of cases in each tract if this overall rate were applied to the tract-level population size.

Finally, calculate the standardized mortality ratio, the number observed divided by the number expected (if the overall rate applied equally everywhere).

```
# Calculate overall rate
rate <- sum(nydata$TRACTCAS) / sum(nydata$POP8)

# Set expected number of cases (population * rate)
Expected <- nydata$POP8 * rate

# Calculate standardized mortality ratio (SMR)
SMR <- nydata$TRACTCAS / nydata$Expected
```

Now fit the ICAR model and output the estimates.

```
# Fit ICAR model
# Assuming here that the order of nydata and nyadjmat are the same
ID_new <- seq(nrow(nydata)) # I think INLA needs the IDs to be 1:n
nydata$ID_new <- ID_new
nyadjlw <- mat2listw(nyadjmat, row.names = ID_new)
listw_NY <- nb2listw(nyadjlw$neighbours, style="B")
# FEED THE FOLLOWING INTO INLA'S graph ARGUMENT instead of listw_NY
nyadj_matrix <- as(nb2mat(nyadjlw$neighbours, style = "B"), "Matrix")

m.icar <- inla(trunc(TRACTCAS) ~ 1 + PCTAGE65P + AVGIDIST + PCTOWNHOME +
              f(ID_new, model = "besag", graph = nyadj_matrix),
              data = as.data.frame(nydata), E = Expected, family = "poisson",
              control.predictor = list(compute = TRUE),
              control.compute = list(dic = TRUE, waic = TRUE))

summary(m.icar)

##
## Call:
##   c("inla(formula = trunc(TRACTCAS) ~ 1 + PCTAGE65P + AVGIDIST +
##     PCTOWNHOME + ", " f(ID_new, model = \"besag\", graph = nyadj_matrix),
##     family = \"poisson\", \", \" data = as.data.frame(nydata), E = Expected,
##     control.compute = list(dic = TRUE, \", \" waic = TRUE), control.predictor
##     = list(compute = TRUE))\" )
## Time used:
##   Pre = 5.11, Running = 1.51, Post = 0.304, Total = 6.92
## Fixed effects:
##           mean      sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept) -0.474 0.173    -0.820   -0.472    -0.141 -0.468  0
## PCTAGE65P    4.182 0.610     2.965    4.189     5.362  4.202  0
## AVGIDIST      0.291 0.083     0.120    0.294     0.446  0.300  0
## PCTOWNHOME   -0.395 0.199    -0.782   -0.397    -0.001 -0.400  0
##
## Random effects:
##   Name      Model
##   ID_new Besags ICAR model
##
## Model hyperparameters:
##           mean      sd 0.025quant 0.5quant 0.975quant   mode
## Precision for ID_new 16201.62 19279.73      6.12  9570.27  68338.34  5.47
##
## Expected number of effective parameters(stddev): 5.32(5.28)
## Number of equivalent replicates : 52.86
##
## Deviance Information Criterion (DIC) .....: 969.21
## Deviance Information Criterion (DIC, saturated) ....: 418.19
## Effective number of parameters .....: 5.81
##
## Watanabe-Akaike information criterion (WAIC) ...: 969.63
## Effective number of parameters .....: 6.59
##
## Marginal log-Likelihood: -698.41
## Posterior marginals for the linear predictor and
```

```
## the fitted values are computed
```