# Model fitting: McMC and INLA

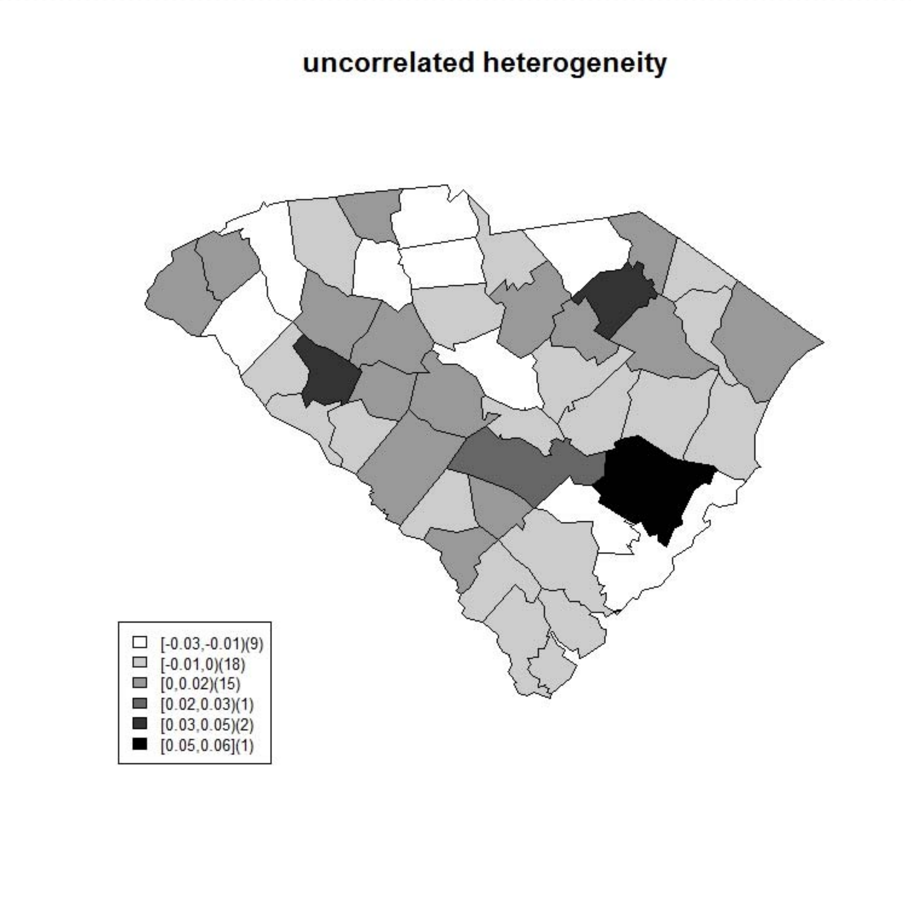
- Conventionally Markov chain Monte Carlo is used to estimate posterior quantities for Bayesian models (such as the convolution or log-normal models)
  - WinBUGS is designed to do this via two basic methods
    - Gibbs sampling
    - Metropolis –Hastings
- Approximation to posterior distributions has recently become available via Laplace approximation in the INLA package
  - Does not require iterative computation (unlike McMC)
  - Fast computation

# INLA results

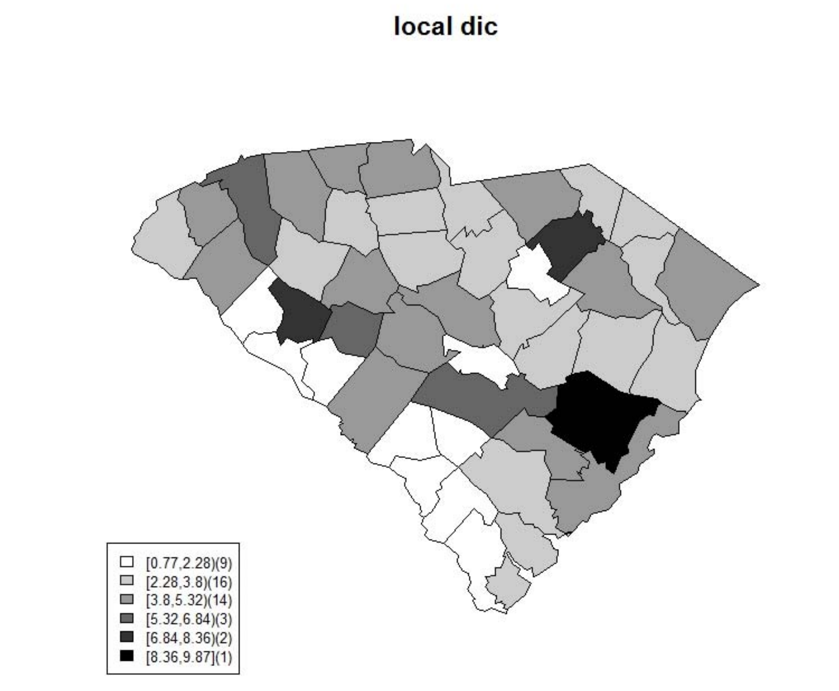
- CODE:

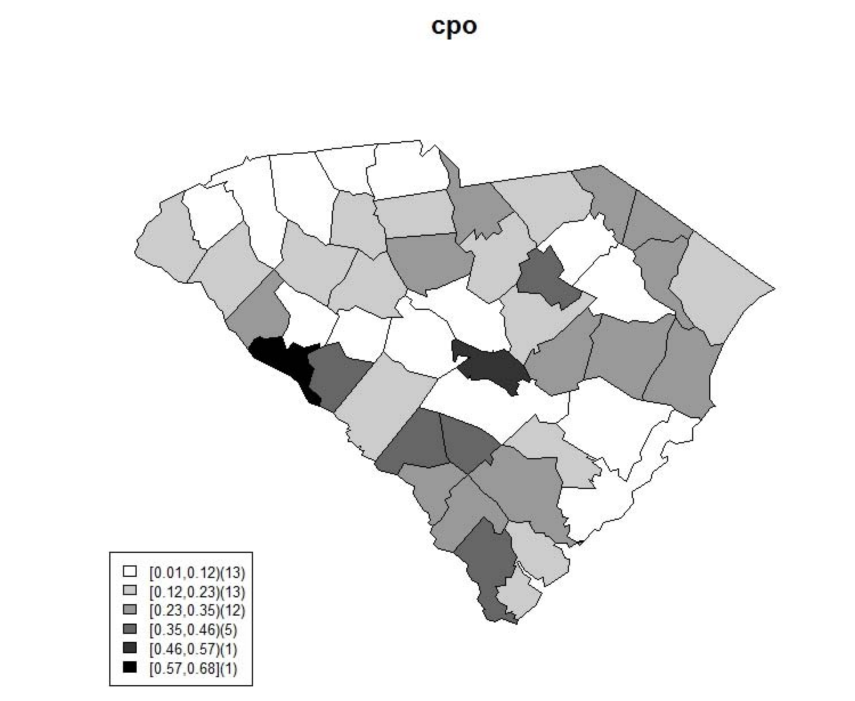
```
formula1.UH = obs ~1+ f(region, model = "iid")

resultUH = inla(formula1.UH,family="poisson",
data=SCcongen90,control.compute=list
(dic=TRUE,cpo=TRUE,graph=TRUE),E=expe)
sum<-resultUH$summary.random
RE1<-sum$region[1:46,2]    # uncorrelated RE
```

# Results: UH component (x 100)



# Results: local DIC and cpo





# INLA

- Integrated Nested Laplace Approximation
  - Posterior approximation which relies on numerical integration and sparse matrix analysis
  - Is particularly suited to Gaussian models, especially if the effects are log-Gaussian or Gaussian
  - Linear Mixed Models  Or
  - Generalized linear mixed models

# INLA more formally

- Laplace approximation matches the mode and curvature of a Gaussian distribution to the posterior in question and uses this to provide an integral approximation to the density.
- For models close to Gaussian then the approximation is good.

# How its computed

$\boldsymbol{y}$ outcome data

$\boldsymbol{\lambda}$ parameters

$\phi$ hyperparameters

$$\mathrm{P}(\boldsymbol{\lambda}_i \mid \boldsymbol{y}) = \int_{\phi} P(\boldsymbol{\lambda}_i \mid \boldsymbol{y}, \phi) P(\phi \mid \boldsymbol{y}) d\phi$$

$$\approx \sum_{k} \Delta_k \overline{P}(\boldsymbol{\lambda}_i \mid \boldsymbol{y}, \phi_k) \overline{P}(\phi_k \mid \boldsymbol{y})$$

*where* $\overline{P}$ denotes the Laplace approximation

# LMM or GLMM

$y_i = x_i^T \beta + z_i^T \gamma + e_i$

or

$E(y_i) = \mu_i$; f(y) non-Gaussian (continuous/discrete)

$g(\mu_i) = x_i^T \beta + z_i^T \gamma$

where x is a desgin matrix of covaroates

and z is a design matrix of random effects

and γ is a unit vector

# Some examples of spatial mixed models

- Geostatistical data:

$$y_i : outcome\ measured\ at\ location\ s_i$$

$$s_i = \{s_1, s_2\}$$

$$\mathbf{y} : N(\boldsymbol{\mu}, \Sigma)$$

$$\Sigma_{ij} = cov(s_i, s_j)$$

$$e.g.$$

$$y_i = x_i^T \beta + e_i;\ \mathbf{e} \sim N(0, \Sigma)$$

# Examples

- Small area health data:

$$
\begin{aligned}
& y_i \sim Pois(\mu_i) \\
& \mu_i = e_i \theta_i \\
& \log(\mu_i) = \log(e_i) + \log(\theta_i) \\
& \log(\theta_i) = x_i^T \beta + z_i^T \gamma
\end{aligned}
$$

# Uncorrelated heterogeneity (UH)

- UH model (random intercept)
  - Uncorrelated Noise model
  - Baseline risk model
  - Assumes no spatial correlation or trend
  - Zero mean Gaussian prior distributions for effects
    - Intercept and random effect

$$z_{1i} \sim N(0,\tau_1^{-1});\ \beta_0 \sim N(0,\tau_0^{-1})$$

# Correlated Heterogeneity (CH)

- This component is where correlation is introduced.
- Some times called *clustering* or *spatially-structured*
- We could assume a variety of forms for spatial correlation
- We often use a CAR model as it is simple and easy to fit
- It can be an improper  or proper prior distribution
- The ICAR (improper) is easy to fit on INLA and WinBUGS
- The ICAR does not have a correlation parameter: the precision controls the variation  and  correlation
- It is adaptive in that the variation depends on the neighborhoods

$$z_{2i} \mid \mathbf{z}_{2j \neq i} \sim N(\overline{z}_{\delta_i}, \tau_1^{-1} / n_{\delta_i})$$

# Models on INLA

- INLA operates as for the LM function on R
  - Two components:
    - formula and inla call

- Example:

```
>formula1=y~1+x
>result1=inla(formula1,
family=“gaussian”,data=‘dataframe’)
```

- This fits a linear regression with intercept between y and x

# INLA basic regression

```
x1<-c(1.1,2.3,3.4,4.5,5.4)
x2<-c(-2.3,4.5,3.6,6.8,12.7)
y<-c(1.2,1.4,2.3,3.2,1.2)
As<-data.frame(x1,x2,y)

library(INLA)
##  1 predictor
formula1<-y~1+x1      # formula for model
res1<-inla(formula1,family="gaussian",data=As,
control.compute=list(dic=TRUE,cpo=TRUE))           #fitting model

summary(res1)      #   displays the summary of the fit
sum1<-res1$summary.fixed    # storing the regression estimates
res1$dic    #  displaying the DIC results
```

# More sophistication

- **Two predictors**

```
formula2<-y~1+x1+x2
res2<-inla(formula2,family="gaussian",data=As,
control.compute=list(dic=TRUE,cpo=TRUE))
```

- **Random effect**

```
## random effect (one predictor and individual level
random effect)
ind<-seq(1:5)
formula3<-y~1+x1+f(ind,model='iid')
```

# Random slope, factor and RW smoothing model

```
ind2<-c(1,1,1,2,2)

## random slope model
Formula5<-y~1+x1+f(ind,x2,model="iid")

# factor random effect
Formula6<-y~1+x1+f(ind2,model="iid")

## smoothed RW model on predictor
Formula7<-y~1+x1+f(x2,model="rw1")
```

# Poisson examples (simple_Poisson_INLA_examples.txt)

```
library(INLA)
# Poisson Bayesian regression

#EITHER:
formula1<-counts~1+POV
res1<-inla(formula1,family="poisson",data=death9,
control.compute=list(dic=TRUE,cpo=TRUE),E=EXPE)
summary(res1)
#OR:

formula1<-counts~1+POV+offset(log(EXPE))
res1<-inla(formula1,family="poisson",data=death9,
control.compute=list(dic=TRUE,cpo=TRUE))
summary(res1)
```

# Random effect models

```
#Poisson random intercept model
ind<-seq(1:9)
formula2<-
counts~1+POV+offset(log(EXPE))+f(ind,model="iid",
param=c(2,1))
res2<-inla(formula2,family="poisson",data=death9,
control.compute=list(dic=TRUE,cpo=TRUE))
summary(res2)
```

# Goodness of Fit

- Goodness-of-fit (GOF) measures should be used to compare how well models fit the data

# Goodness of Fit

- Various measures are available:

Deviance:

$$D(\boldsymbol{\theta}) = -2\sum_{i=1}^{m} log\, f(y_i, \boldsymbol{\theta}) = -\,2l(\mathbf{y}, \boldsymbol{\theta})$$

Mean square error (or other residual based measures)

MSE:

$$MSE_y = \frac{1}{m}\sum_{i=1}^{m}\{y_i - \hat{y}_i\}^2$$

*where* $\hat{y}_i$ could be a fitted value

Fitted values could be computed in different ways

AIC, BIC, DIC: information criterion measures that penalize for number of parameters

# Deviance Information Criterion

- DIC is defined as:

$$DIC = \bar{D}(\boldsymbol{\theta}) + pD$$

$$pD = \bar{D}(\boldsymbol{\theta}) - \hat{D}(\boldsymbol{\theta}) = dbar - dhat$$

$\hat{D}(\boldsymbol{\theta})$: deviance evaluated at posterior estimates of $\boldsymbol{\theta}$

$\bar{D}(\boldsymbol{\theta})$: deviance averaged over sample

# Deviance Information Criterion

- Deviance information Criterion (DIC)
  - Smaller better; comparative tool
  - pD measures the effective number of parameters
  - Hence DIC and pD should be small ideally
  - Relative measure only: can have negative DIC ........ BUT cant have negative pD

  - Difference of 3-5 in DIC is ‘significant’ when you fit a series of models

# CPO and Marginal Predictive Likelihood (BDM2013 ch 4)

- Conditional predictive ordinate (CPO)
  - Can be computed from INLA or WinBUGS output for each observation

$$CPO_i^{-1} = \left\{ \frac{1}{G} \sum_{g=1}^{G} L_i(\boldsymbol{\theta}^g)^{-1} \right\}$$

  - Marginal predictive likelihood is a measure of overall GOF

$$M = \sum_i \log(CPO_i)$$

©Andrew B Lawson 2017

# Small Area Health models on INLA

- Assume a Poisson likelihood and $\mu_i = e_i\theta_i$
- and $$\log(\mu_i) = \underbrace{\log(e_i)}_{offset} + \log(\theta_i)$$
- Linear modeling of the log relative risk
- Usually we assume an additive model for effects. eg

$$\log(\theta_i) = \alpha_0 + t_i + z_{1i} + z_{2i}$$
$$t_i = \mathbf{x}_i^T \boldsymbol{\alpha}$$

# R-INLA demo

- INLA basics
- Graphs in R

# Thematic Mapping in R

- Polygon objects can be handled in R
  - library(maptools)

- Imported from shapefiles:

```
>polys<-readShapePoly("filename.shp")
>plot(polys)
```

- Read in from GeoBUGS export files:

# R mapping

- Reading in polygons from shapefile

```
>setwd(“working directory")
>polySC<-readShapePoly("co45_d00.shp")
>plot(polySC)
```

- Reading in from GeoBUGS export file

```
>polySC<-readSplus("SC_geobugsSPlus.txt")
>plot(polySC)
```

- polySC is a polygon object and can be used for mapping on R.

# Special INLA format

- INLA requires that a special format is used when fitting models with spatial components.
  - This format can be obtained via two routes:

1) if adj and num vectors are available (already read into R) then the command

```
>geobugs2inla(adj, num, graph.file="SC_poly.txt")
```

will create a valid spatial graph file for inla models

# Special INLA format

2) using library spdep

```
>library(spdep)
>adjpoly<-poly2nb(polySC)
>nb2INLA("SC_poly.txt",adjpoly)
```

will create the graph file "SC_poly.txt"

# General Thematic Plotting on R

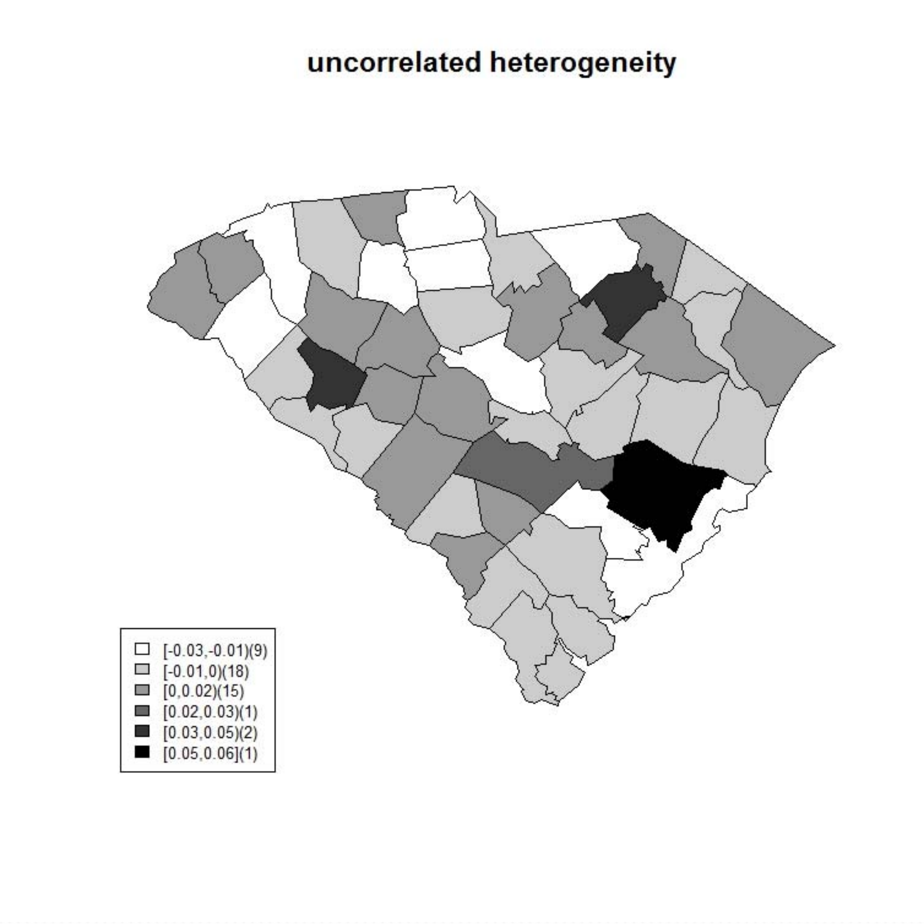
- Using ‘fillmap’ allows vectors to be mapped on R

```
>source(“fillmap.R”)
```

- a model fit is obtained in 'result1'
- then we can map result1$summary.random[,2] by the following:

```
>rand<-result1$summary.random$region[,2]
>fillmap(polySC,"random effect", rand*100, n.col=6)
```

# Example plot



# Some INLA Examples

- UH model
- CH model (ICAR)
- Convolution

- SCcongen_INLA_models.txt

# INLA Examples

- Fit a UH model to the SC congen90 data
  - Plot of UH effect
  - DIC and local DIC map
- Fit a CH model to the same data
  - Plot of CH component
  - DIC and local DIC map
- Fit a convolution model to the same data
  - Plot of UH and CH components
  - DIC and local DIC map

# Data input

```
SCcongen90<-
list(obs=c(0,7,1,5,1,1,5,16,0,17,4,0,0,1,1,7,1,3,0,0,8,2,13,7,0,8,
0,3,2,4,1,11,0,1,2,3,3,8,6,14,3,11,6,0,1,5),
expe=c(1.0807,6.3775,0.622,6.6854,0.9142,1.0744,5.6518,8.
1682,0.5749,18.0989,2.174,1.6619,1.9321,1.6148,1.6713,
3.0819,1.7562,4.9952,0.9362,1.2001,6.1293,2.5604,15.8589,2
.9437,1.0399,7.276,0.9739,2.064,2.7206,2.8275,
0.9425,8.828,0.3644,1.775,1.5111,1.5111,2.5321,4.5836,3.9647,
15.0264,0.732,10.8292,5.9848,1.4357,1.9949,6.9807))
region<-seq(1:46)
```

# UH model Commands

```
>library(INLA)
>formulaUH = obs~ f(region, model = "iid")
>resultUH = inla(formulaUH,family="poisson",
data=SCcongen90,control.compute=list(dic=TRUE),E=e
xpe)

>summary(resultUH)
>resultUH$summary.fixed;resultUH$summary.random
>rand<-resultUH$summary.random$region[,2]
```

# Graphics

```
>fillmap(polySC,"random effect",rand*100,n.col=6)
>dic<-resultUH$dic$local.dic
>fillmap(polySC,”DIC”,dic,n.col=6)
```

# CH models 1  ICAR

```
>formulaCH = obs~f(region,
model="besag",graph="SC.graph“)
>resultCH =
inla(formulaCH,family="poisson",data=SCcongen90,con
trol.compute=list(dic=TRUE,cpo=TRUE,graph=TRUE),E
=expe)
```

# CH model 2: Convolution

```
>region2<-region
>formulaCONV1 = obs ~
f(region,model="iid",)+f(region2, model = "besag", graph
= "SC.graph“)
>result1 = inla(formulaCONV1,family="poisson",
data=SCcongen90,control.compute=list(dic=TRUE,grap
h=TRUE),E=expe)
```

# CH models Conv alternative

```
>formulaCONV2 = obs ~ f(region, model = "bym", graph = "SC.graph“)
>result1 = inla(formulaCONV2,family="poisson",
data=SCcongen90,control.compute=list(dic=TRUE,graph=TRUE),E=expe)
```

# INLA(Sccongen_INLA_models.txt)

- SC county level congenital abnormality deaths 1990
- UH, CH and % under poverty covariate

DIC/ML results:

| Model | DIC | pD | ML |
|---|---|---|---|
| UH only | 169.35 | 1.017 | -85.46 |
| UH + POV | 171.36 | 2.01 | -92.96 |
| UH+CH+POV | 171.36 | 2.03 | -124.37 |

# Finally

- I have given an overview of the use of INLA for Bayesian disease mapping
- For a range of basic mapping tasks INLA is very fast and flexible
- Can also be used for space-time models
- In addition a whole range of other spatial and non-spatial models are available : longitudinal, survival, imaging, econometrics, point process modeling
- There is also WinBUGS, OpenBUGS, JAGS and STAN

# Book

- Blangiardo, M. and Cameletti, M. (2015)

Spatial and Spatio-temporal Bayesian Models with R – INLA, Wiley, New York