SISMID Spatial Statistics in Epidemiology and Public Health
2015 R Notes: Cluster Detection and Clustering for Count Data

Jon Wakefield
Departments of Statistics and Biostatistics, University of Washington

2015-07-18

North Carolina SIDS Data

The `nc.sids` data frame has 100 rows and 21 columns and can be found in the `spdep` library.

It contains data given in Cressie (1991, pp. 386-9), Cressie and Read (1985) and Cressie and Chan (1989) on sudden infant deaths in North Carolina for 1974–78 and 1979–84.

The data set also contains the neighbour list given by Cressie and Chan (1989) omitting self-neighbours (`ncCC89.nb`), and the neighbour list given by Cressie and Read (1985) for contiguities (`ncCR85.nb`).

Data is available on the numbers of cases and on the number of births, both dichotomized by a binary indicator of race.

The data are ordered by county ID number, not alphabetically as in the source tables.

North Carolina SIDS Data

The code below plots the county boundaries along with the observed SMRs.

The expected numbers are based on internal standardization with a single stratum.

```r
library(maptools)
nc.sids <- readShapePoly(system.file("etc/shapes/sids.shp",
    package = "spdep")[1], ID = "FIPSNO",
    proj4string = CRS("+proj=longlat +ellps=clrk66"))
nc.sids2 <- nc.sids  # Create a copy, to add to
Y <- nc.sids$SID74
E <- nc.sids$BIR74 * sum(Y)/sum(nc.sids$BIR74)
nc.sids2$SMR74 <- Y/E
nc.sids2$EXP74 <- E
brks <- seq(0, 5, 1)
```

SMR Plot

We map the SMRs, and see a number of counties with high relative risks.

```
spplot(nc.sids2, "SMR74", at = brks,
    col.regions = grey.colors(5, start = 0.9,
        end = 0.1))
```

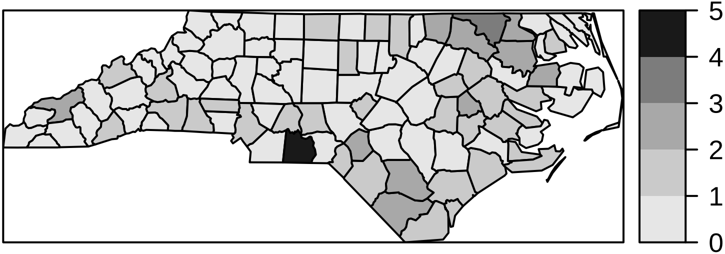


Figure 1: Map of SMRs for SIDS in 1974 in North Carolina

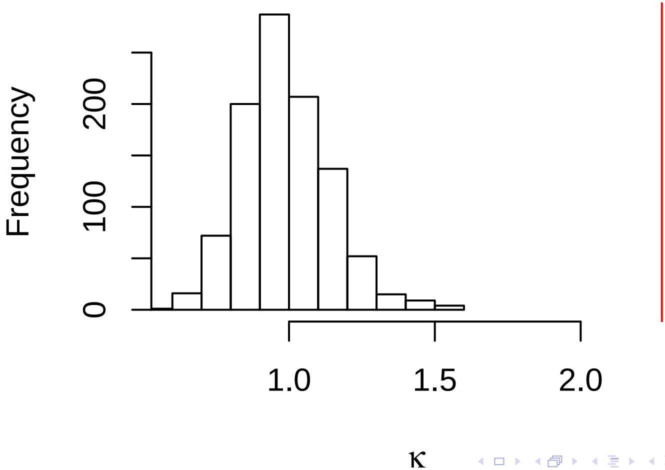
Overdispersion

Examine κ, the overdispersion statistic, and use a Monte Carlo test to examine significance.

```r
library(spdep)
kappaval <- function(Y, fitted, df) {
    sum((Y - fitted)^2/fitted)/df
}
mod <- glm(Y ~ 1, offset = log(E), family = "quasipoisson")
kappaest <- kappaval(Y, mod$fitted, mod$df.resid)
nMC <- 1000
ncts <- length(E)
yMC <- matrix(rpois(n = nMC * ncts, lambda = E),
    nrow = ncts, ncol = nMC)
kappaMC <- NULL
for (i in 1:nMC) {
    modMC <- glm(yMC[, i] ~ 1, offset = log(E),
        family = "quasipoisson")
    kappaMC[i] <- kappaval(yMC[, i], modMC$fitted,
        modMC$df.resid)
}
```

Overdispersion

```
hist(kappaMC, xlim = c(min(kappaMC),
    max(kappaMC, kappaest)), main = "",
    xlab = expression(kappa))
abline(v = kappaest, col = "red")
```



Disease Mapping

We first fit a non-spatial random effects model:

$$\begin{aligned} Y_i|\alpha, V_i &\sim_{iid} \text{Poisson}(E_i e^{\alpha + V_i}), \\ V_i|\sigma_v^2 &\sim_{iid} N(0, \sigma_v^2) \end{aligned}$$

```r
library(INLA)
nc.sids2$ID <- 1:100
m0 <- inla(SID74 ~ f(ID, model = "iid"),
    family = "poisson", E = EXP74, data = as.data.frame(nc.sids2
    control.predictor = list(compute = TRUE))
```

Disease Mapping

Examine the first few “fitted values”, summaries of the posterior distribution of $\exp(\alpha + V_i)$, $\qquad i = 1, \ldots, n$.

```
head(m0$summary.fitted.values)
##                            mean        sd 0.025quant  0.5quant 0.975quant
## fitted.predictor.001 1.2515021 0.2930181  0.7548490 1.2250844   1.899824
## fitted.predictor.002 0.7665958 0.2700582  0.3481650 0.7299039   1.397177
## fitted.predictor.003 0.9149708 0.3494437  0.3989681 0.8598644   1.751025
## fitted.predictor.004 2.7309425 0.7626511  1.5074088 2.6400575   4.470065
## fitted.predictor.005 0.9027425 0.3177245  0.4165809 0.8575336   1.650257
## fitted.predictor.006 0.8544442 0.3152039  0.3789292 0.8076757   1.601193
##                            mode
## fitted.predictor.001 1.1747221
## fitted.predictor.002 0.6631748
## fitted.predictor.003 0.7637463
## fitted.predictor.004 2.4583333
## fitted.predictor.005 0.7763712
## fitted.predictor.006 0.7245109
```

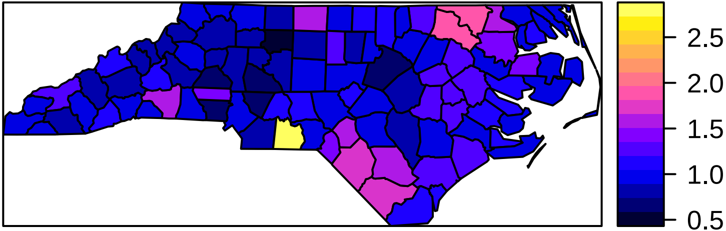
Disease Mapping

Create two interesting inferential summaries.

```
# We add the posterior mean of the
# relative risk to our object
nc.sids2$RRpmean0 <- m0$summary.fitted.values[,
    1]
# Also, a binary indicator of whether
# posterior median is greater than 1.5
# (an epidemiologically significant
# value)
nc.sids2$RRind0 <- m0$summary.fitted.values[,
    4] > 1.5
```

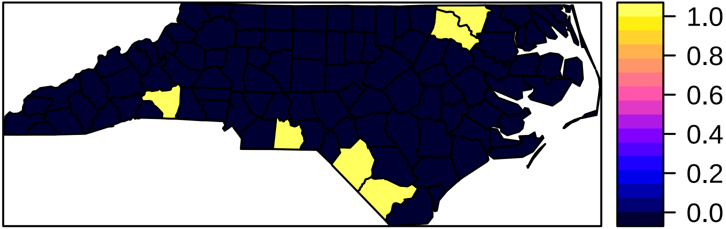
Disease Mapping

```
# Display relative risk estimates
spplot(nc.sids2, "RRpmean0")
```



Disease Mapping

```
# Display indicators of whether 0.5
# points above 1.5
spplot(nc.sids2, "RRind0")
```



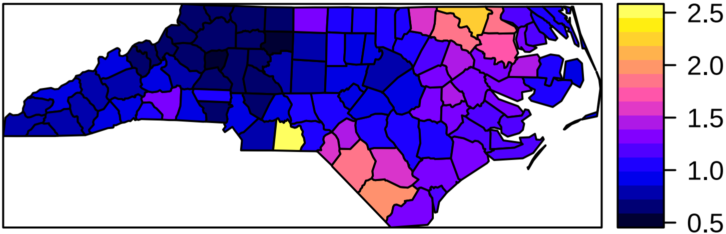
Disease Mapping

We now fit a model with non-spatial and spatial random effects.

```r
# nc.sids2 <- readShapePoly(
# system.file('etc/shapes/sids.shp',
# package='spdep')[1]) Create adjacency matrix
# nc.nb <- poly2nb(nc.sids)
nc.sids2$ID2 <- 1:100
m1 <- inla(SID74 ~ 1 + f(ID, model = "iid") + f(ID2,
    model = "besag", graph = "examples/NC.graph"),
    family = "poisson", E = EXP74, data = as.data.frame(nc.sids2
    control.predictor = list(compute = TRUE))
nc.sids2$RRpmean1 <- m1$summary.fitted.values[, 1]
nc.sids2$RRind1 <- m1$summary.fitted.values[, 4] >
    1.5
```

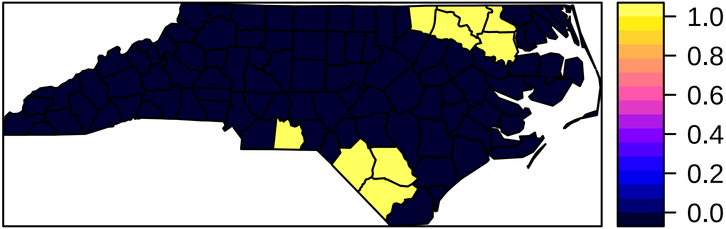
Disease Mapping

```
# Display
spplot(nc.sids2, "RRpmean1")
```



Disease Mapping

```r
# Display areas with medians above 1.5, ie those
# areas with greater than 50% chance of exceedence
# of 1.5.
spplot(nc.sids2, "RRind1")
```



Disease Mapping: Comparison of posterior means

```r
plot(nc.sids2$RRpmean1 ~ nc.sids2$RRpmean0,
    type = "n", xlab = "Non-spatial model",
    ylab = "Spatial model")
text(nc.sids2$RRpmean1 ~ nc.sids2$RRpmean0)
abline(0, 1)
```

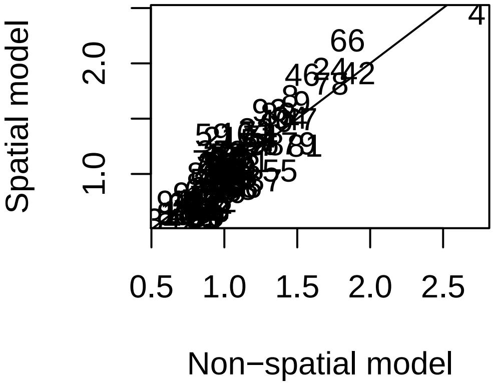


Figure 3:

Disease Mapping

We now examine the variances of the spatial and non-spatial random effects.

Recall that the ICAR model variance has a conditional interpretation.

To obtain a rough estimate of the marginal variance we obtain the posterior median of the U_i's and evaluate their variance.

From the output below, we conclude that the spatial random effects dominate for the SIDS data so that we conclude there is clustering of cases in neighboring areas.

Disease Mapping

```
# Extract spatial random effects
U <- m1$summary.random$ID2[5]
sqrt(var(U))  # 0.33
##          0.5quant
## 0.5quant 0.3314246
# variance of non-spatial
m1$summary.hyperpar
##                          mean           sd  0.025quant    0.5quant
## Precision for ID  17946.354826 1.751483e+04 1204.781841 12805.63061
## Precision for ID2     2.299427 8.843653e-01    1.091817     2.12514
##                    0.975quant        mode
## Precision for ID  64499.89990 3275.258473
## Precision for ID2     4.50051    1.823358
```

Clustering via Moran's I

We evaluate Moran's test for spatial autocorrelation using the "W" style weight function: this standardizes the weights so that for each area the weights sum to 1.

To obtain a variable with approximately constant variance we form residuals from an intercept only model.

```r
library(spdep)
data(nc.sids)
col.W <- nb2listw(ncCR85.nb, style = "W", zero.policy = TRUE)
quasipmod <- glm(SID74 ~ 1, offset = log(EXP74), data = nc.sids2,
    family = quasipoisson())
sidsres <- residuals(quasipmod, type = "pearson")
```

Clustering via Moran's *I*

```
moran.test(sidsres, col.W)
##
##  Moran's I test under randomisation
##
## data:  sidsres
## weights: col.W
##
## Moran I statistic standard deviate = 2.4351, p-value = 0.007444
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.147531140      -0.010101010       0.004190361
```

Clustering via Moran's *I*

Moran's test may suggest spatial autocorrelation if there exists a non-constant mean function.

Below we fit a model with Eastings and Northings (of the County seat) as covariates – both show some association and the significance of the Moran statistic is reduced, though still significant.

```
quasipmod2 <- glm(SID74 ~ east + north, offset = log(EXP74),
    data = nc.sids2, family = quasipoisson())
summary(quasipmod2)
##
## Call:
## glm(formula = SID74 ~ east + north, family = quasipoisson(),
##     data = nc.sids2, offset = log(EXP74))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7961  -1.0249  -0.3475   0.6043   4.7261
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.2465437  0.2680159  -0.920  0.35992
## east         0.0020105  0.0006469   3.108  0.00247 **
## north       -0.0028032  0.0014545  -1.927  0.05687 .
## ---
```

North Carolina SIDS Data: Disease Mapping

```
par(mar = c(0.1, 0.1, 0.1, 0.1))
spplot(nc.sids2, "res")
```

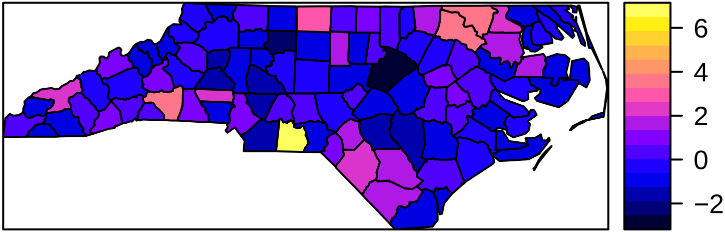


Figure 4:

Clustering via Moran's *I*

```
moran.test(sidsres2, col.W)
##
##  Moran's I test under randomisation
##
## data:  sidsres2
## weights: col.W
##
## Moran I statistic standard deviate = 2.1328, p-value = 0.01647
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.127428361      -0.010101010       0.004157993
```

Clustering via Geary's *c*

We now use Geary's statistic on the detrended residuals, and come to the same conclusion

```
geary.test(sidsres2, col.W)
##
##  Geary's C test under randomisation
##
## data:  sidsres2
## weights: col.W
##
## Geary C statistic standard deviate = 2.3479, p-value = 0.009439
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic       Expectation          Variance
##         0.8195420         1.0000000         0.0059072
```

Clustering via Moran's *I*

We now use Moran's statistic on the detrended residuals, but with the binary "B" weight option.

This option has unstandardized weights.

Note the asymmetry in the "W" weights option in the figure below.

The conclusion, evidence of spatial autocorrelation, is the same as with the standardized weights option.

```
col.B <- nb2listw(ncCR85.nb, style = "B", zero.policy = TRUE)
moran.test(sidsres2, col.B)
##
##  Moran's I test under randomisation
##
## data:  sidsres2
## weights: col.B
##
## Moran I statistic standard deviate = 2.2357, p-value = 0.01269
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.125344196      -0.010101010       0.003670354
```

Clustering via Moran's *I*

We now use Moran's statistic on the detrended residuals, but with the binary “B" weight option.

This option has unstandardized weights.

Note the asymmetry in the “W" weights option in the figure below.

The conclusion, evidence of spatial autocorrelation, is the same as with the standardized weights option.

```
col.B <- nb2listw(ncCR85.nb, style = "B", zero.policy = TRUE)
moran.test(sidsres2, col.B)
##
##  Moran's I test under randomisation
##
## data:  sidsres2
## weights: col.B
##
## Moran I statistic standard deviate = 2.2357, p-value = 0.01269
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.125344196      -0.010101010       0.003670354
```

Neighborhood Options

```r
library(RColorBrewer)
pal <- brewer.pal(9, "Reds")
par(mfrow = c(1, 2))
z <- t(listw2mat(col.W))
brks <- c(0, 0.1, 0.143, 0.167, 0.2, 0.5, 1)
nbr3 <- length(brks) - 3
image(1:100, 1:100, z[, ncol(z):1], breaks = brks,
    col = pal[c(1, (9 - nbr3):9)], main = "W style",
    axes = FALSE)
box()
z <- t(listw2mat(col.B))
brks <- c(0, 0.1, 0.143, 0.167, 0.2, 0.5, 1)
nbr3 <- length(brks) - 3
image(1:100, 1:100, z[, ncol(z):1], breaks = brks,
    col = pal[c(1, (9 - nbr3):9)], main = "B style",
    axes = FALSE)
box()
```

Neighborhood Options

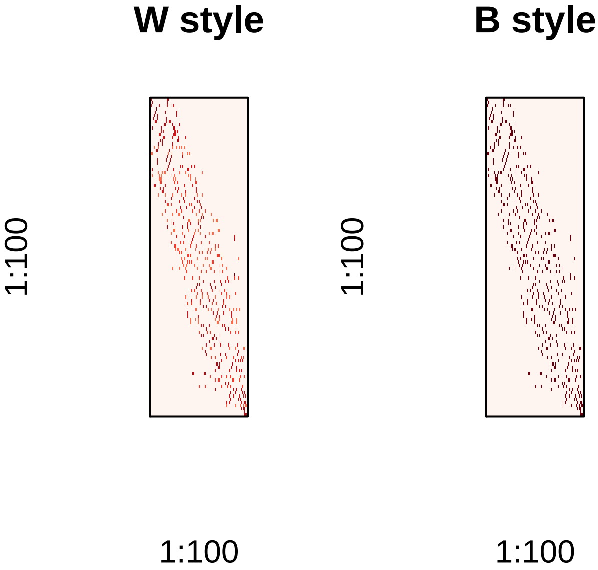


Figure 5:

North Carolina SIDS Data: Conclusions

Both of the Moran's I and Geary's c methods suggest that there is evidence of clustering in these data.

North Carolina SIDS Data: Clustering

We implement Openshaw's method using the centroids of the areas in data.

Circles of radius 30 are used and the centers are placed on a grid of size 10.

For multiple radii, multiple calls are required.

The significance level for calling a cluster is 0.002.

```r
library(spdep)
data(nc.sids)
sids <- data.frame(Observed = nc.sids$SID74)
sids <- cbind(sids, Expected = nc.sids$BIR74 * sum(nc.sids$SID74)/sum(nc.sids$B
sids <- cbind(sids, x = nc.sids$x, y = nc.sids$y)
# GAM
library(DCluster)
sidsgam <- opgam(data = sids, radius = 30, step = 10,
    alpha = 0.002)
```

North Carolina SIDS Data

```
plot(sids$x, sids$y, xlab = "Easting", ylab = "Northing")
# Plot points marked as clusters
points(sidsgam$x, sidsgam$y, col = "red", pch = "*")
```

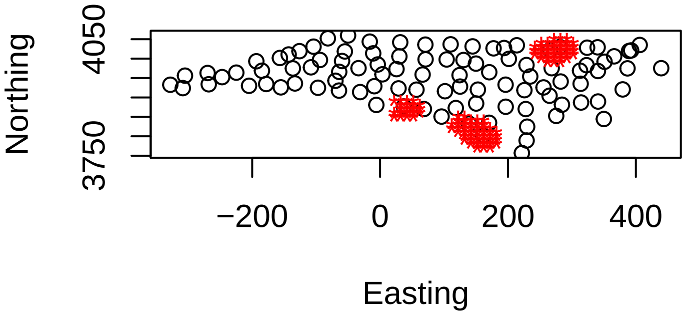


Figure 6:

Clustering via Openshaw

Openshaw results.

```
sidsgam
##          x       y statistic cluster       pvalue size
## 1   151.96 3776.92        15       1 1.743356e-03    1
## 2   161.96 3776.92        15       1 1.743356e-03    1
## 3   171.96 3776.92        15       1 1.743356e-03    1
## 4   141.96 3786.92        15       1 1.743356e-03    1
## 5   151.96 3786.92        15       1 1.743356e-03    1
## 6   161.96 3786.92        15       1 1.743356e-03    1
## 7   171.96 3786.92        15       1 1.743356e-03    1
## 8   181.96 3786.92        15       1 1.743356e-03    1
## 9   131.96 3796.92        15       1 1.743356e-03    1
## 10  141.96 3796.92        15       1 1.743356e-03    1
## 11  151.96 3796.92        15       1 1.743356e-03    1
## 12  161.96 3796.92        15       1 1.743356e-03    1
## 13  171.96 3796.92        15       1 1.743356e-03    1
## 14  181.96 3796.92        15       1 1.743356e-03    1
## 15  131.96 3806.92        46       1 5.531787e-06    2
## 16  141.96 3806.92        46       1 5.531787e-06    2
## 17  151.96 3806.92        46       1 5.531787e-06    2
## 18  161.96 3806.92        23       1 2.042224e-04    2
## 19  171.96 3806.92        23       1 2.042224e-04    2
## 20  181.96 3806.92        15       1 1.743356e-03    1
## 21  121.96 3816.92        31       1 2.612008e-04    1
## 22  131.96 3816.92        31       1 2.612008e-04    1
```

North Carolina SIDS Data: Besag and Newell $k = 20$

```r
library(SpatialEpi)
library(maptools)
library(spdep)
library(maps)
library(ggplot2)
library(sp)
nc.sids <- readShapePoly(system.file("etc/shapes/sids.shp",
    package = "spdep")[1], ID = "FIPSNO", proj4string = CRS("+proj=longlat +ell
referencep <- sum(nc.sids$SID74)/sum(nc.sids$BIR74)
population <- nc.sids$BIR74
cases <- nc.sids$SID74
E <- nc.sids$BIR74 * referencep
SMR <- cases/E
n <- length(cases)
```

North Carolina SIDS Data: Besag and Newell $k = 20$

```r
getLabelPoint <- function(county) {
    Polygon(county[c("long", "lat")])@labpt
}
df <- map_data("county", "north carolina")  # NC region county data
centNC <- by(df, df$subregion, getLabelPoint)  # Returns list
centNC <- do.call("rbind.data.frame", centNC)  # Convert to Data Frame
names(centNC) <- c("long", "lat")  # Appropriate Header
centroids <- matrix(0, nrow = n, ncol = 2)
for (i in 1:n) {
    centroids[i, ] <- c(centNC$lat[i], centNC$long[i])
}
colnames(centroids) <- c("x", "y")
rownames(centroids) <- 1:n
```

North Carolina SIDS Data: Besag and Newell $k = 20$

```r
NCTemp <- map("county", "north carolina", fill = TRUE,
    plot = FALSE)
NCIDs <- substr(NCTemp$names, 1 + nchar("north carolina,"),
    nchar(NCTemp$names))
NC <- map2SpatialPolygons(NCTemp, IDs = NCIDs, proj4string = CRS("+proj=longlat"
# Fix currituck county which is 3 islands
index <- match(c("currituck:knotts", "currituck:main",
    "currituck:spit"), NCIDs)
currituck <- list()
for (i in c(27:29)) currituck <- c(currituck, list(Polygon(NC@polygons[[i]]@Poly
currituck <- Polygons(currituck, ID = "currituck")
```

North Carolina SIDS Data: Besag and Newell $k = 20$

```r
# make new spatial polygons object
NC.new <- NC@polygons[1:(index[1] - 1)]
NC.new <- c(NC.new, currituck)
NC.new <- c(NC.new, NC@polygons[(index[3] + 1):length(NC@polygons)])
NC.new <- SpatialPolygons(NC.new, proj4string = CRS("+proj=longlat"))
NCIDs <- c(NCIDs[1:(index[1] - 1)], "currituck", NCIDs[(index[3] +
    1):length(NC@polygons)])
NC <- NC.new

# SANITY CHECK: Reorder Spatial Polygons of list to
# match order of county
names <- rep("", 100)
for (i in 1:length(NC@polygons)) names[i] <- NC@polygons[[i]]@ID
identical(names, NCIDs)
## [1] FALSE

index <- match(NCIDs, names)
NC@polygons <- NC@polygons[index]
rm(index)

names <- rep("", 100)
for (i in 1:length(NC@polygons)) names[i] <- NC@polygons[[i]]@ID
identical(names, NCIDs)
## [1] TRUE
```

North Carolina SIDS Data: Besag and Newell $k = 20$

```
k <- 20
alpha.level <- 0.01
geo <- centroids
BNresults <- besag_newell(geo, population, cases, expected.cases = NULL,
    k, alpha.level)
BNsig <- length(BNresults$p.values[BNresults$p.values <
    alpha.level])
cat("No of sig results = ", BNsig, "\n")
## No of sig results =  11
resmat <- matrix(NA, nrow = BNsig, ncol = 100)
reslen <- NULL
for (i in 1:length(BNresults$clusters)) {
    reslen[i] <- length(BNresults$clusters[[i]]$location.IDs.included)
    resmat[i, 1:reslen[i]] <- BNresults$clusters[[i]]$location.IDs.included
}
```

North Carolina SIDS Data

```
par(mfrow = c(3, 3), mar = c(0.1, 0.1, 0.1, 0.1))
for (i in 1:6) {
    plot(NC.new)
    plot(NC.new[resmat[i, c(1:reslen[i])]], col = "red",
        add = T)
}
```

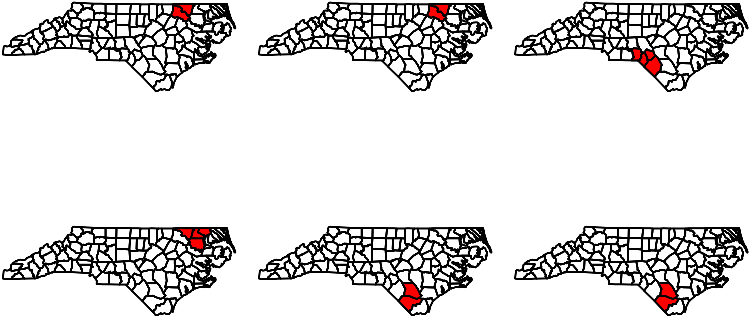


Figure 7:

North Carolina SIDS Data

```
par(mfrow = c(3, 3), mar = c(0.1, 0.1, 0.1, 0.1))
for (i in 6:10) {
    plot(NC.new)
    plot(NC.new[resmat[i, c(1:reslen[i])]], col = "red",
        add = T)
}
```

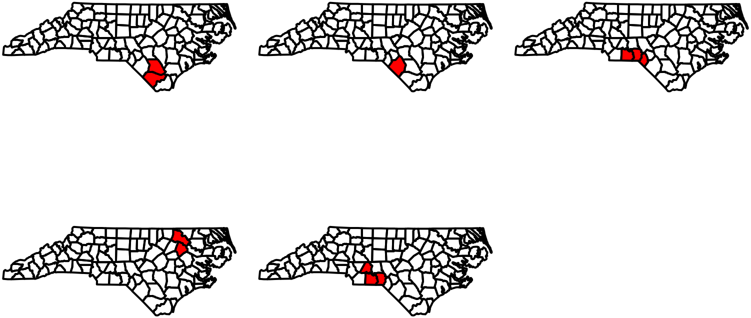


Figure 8:

North Carolina SIDS Data: Besag and Newell $k = 20$

```r
# Kulldorff
pop.upper.bound <- 0.2
n.simulations <- 999
alpha.level <- 0.05
Kpoisson <- kulldorff(geo, cases, population, expected.cases = NULL,
    pop.upper.bound, n.simulations, alpha.level, plot = T)
Kcluster <- Kpoisson$most.likely.cluster$location.IDs.included
```

North Carolina SIDS Data: Besag and Newell $k = 20$

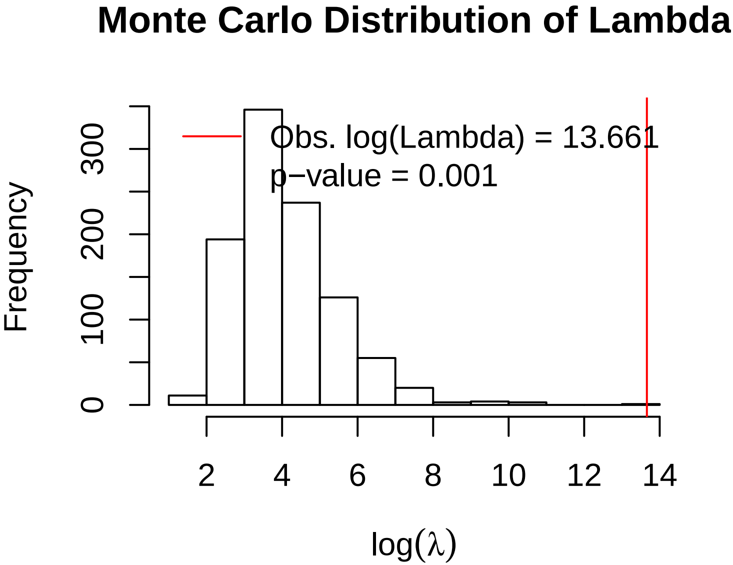
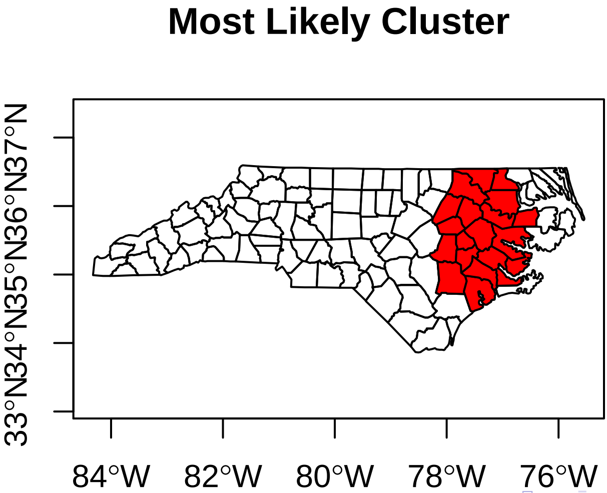


Figure 9:

North Carolina SIDS Data

```r
plot(NC.new, axes = TRUE)
plot(NC.new[Kcluster], add = TRUE, col = "red")
title("Most Likely Cluster")
```



North Carolina SIDS Data: Besag and Newell $k = 20$

Now look at secondary clusters.

Two are significant, and indicated in Figures below

```r
K2cluster <- Kpoisson$secondary.clusters[[1]]$location.IDs.included
plot(NC.new, axes = TRUE)
plot(NC.new[K2cluster], add = TRUE, col = "red")
title("2nd Most Likely Cluster")
```

North Carolina SIDS Data

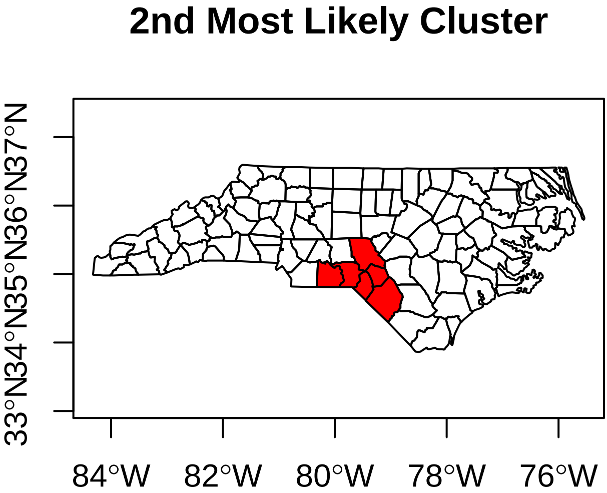


Figure 11:

North Carolina SIDS Data: Bayes cluster model

```
library(spdep)
devtools::install_github("rudeboybert/SpatialEpi")
library(SpatialEpi)
data("nc.sids")
# Load NC map and obtain geographic centroids
library(maptools)
sp.obj <- readShapePoly(system.file("etc/shapes/sids.shp",
    package = "spdep")[1], ID = "FIPSNO", proj4string = CRS("+proj=longlat +ell
centroids <- latlong2grid(coordinates(sp.obj))
```

North Carolina SIDS Data: Bayes cluster model

```
library(maptools)
y <- nc.sids$SID74
population <- nc.sids$BIR74
E <- expected(population, y, 1)
max.prop <- 0.15
k <- 5e-05
shape <- c(2976.3, 2.31)
rate <- c(2977.3, 1.31)
J <- 7
pi0 <- 0.95
n.sim.lambda <- 0.5 * 10^4
n.sim.prior <- 0.5 * 10^4
n.sim.post <- 0.5 * 10^5
output <- bayes_cluster(y, E, population, sp.obj, centroids,
    max.prop, shape, rate, J, pi0, n.sim.lambda, n.sim.prior,
    n.sim.post)
## [1] "Algorithm started on: Sat Jul 18 11:03:27 2015"
## [1] "Geographic objects creation complete on: Sat Jul 18 11:03:28 2015"
## [1] "Importance sampling of lambda complete on: Sat Jul 18 11:03:30 2015"
## [1] "Prior map MCMC complete on: Sat Jul 18 11:03:32 2015"
## [1] "Posterior estimation complete on: Sat Jul 18 11:04:27 2015"
```

Bayesian cluster model

```r
SMR <- y/E
plotmap(SMR, sp.obj, nclr = 6, location = "bottomleft")
plotmap(output$prior.map$high.area, sp.obj, nclr = 6,
    location = "bottomleft")
plotmap(output$post.map$high.area, sp.obj, nclr = 6,
    location = "bottomleft")
barplot(output$pj.y, names.arg = 0:J, xlab = "j", ylab = "P(j|y)
plotmap(output$post.map$RR.est.area, sp.obj, log = TRUE,
    nclr = 6, location = "bottomleft")
```

Bayesian cluster model

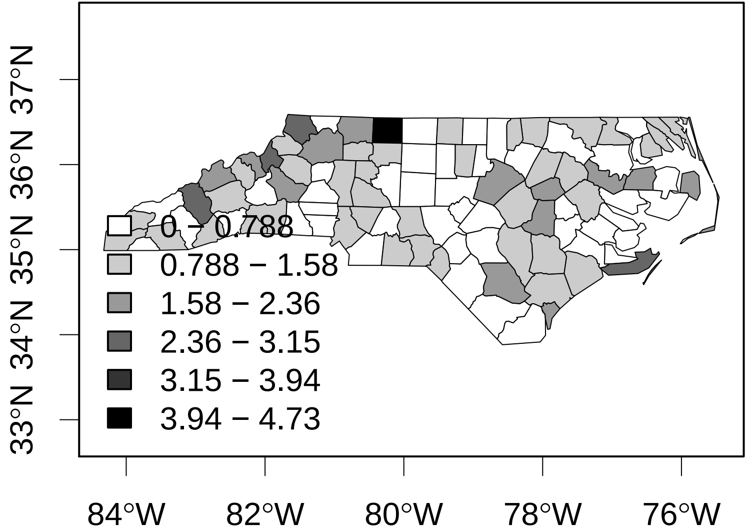


Figure 12: SMRs

Bayesian cluster model

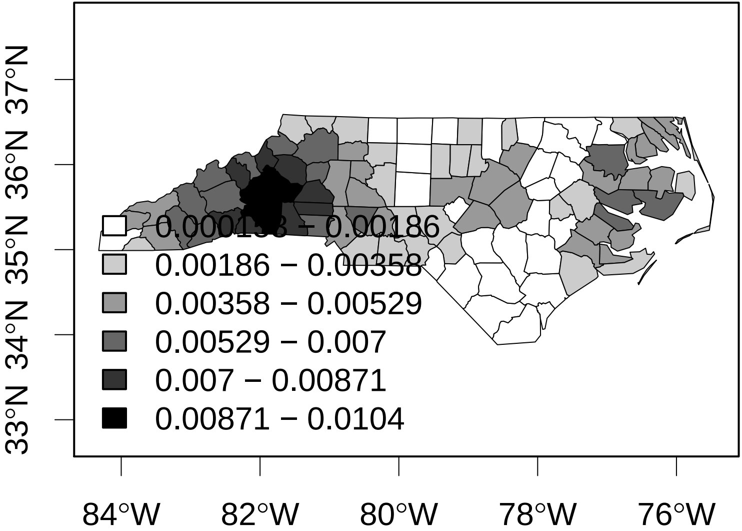


Figure 13: Prior probabilities of lying in a cluster

Bayesian cluster model

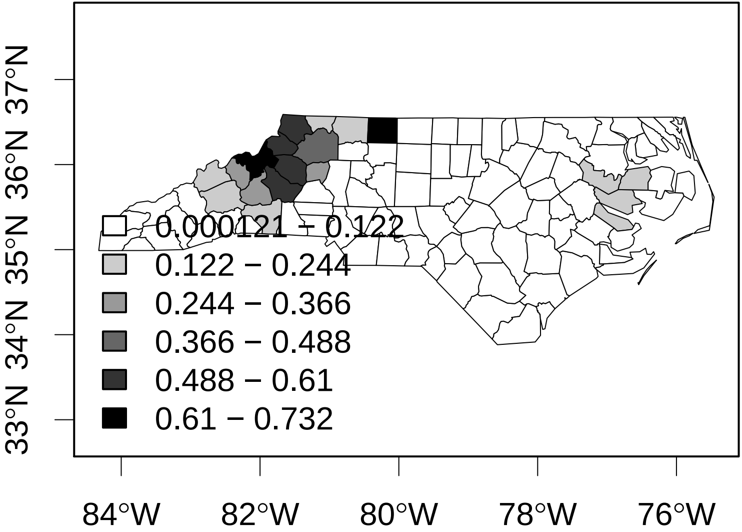


Figure 14: Posterior probability of a cluster

Bayesian cluster model

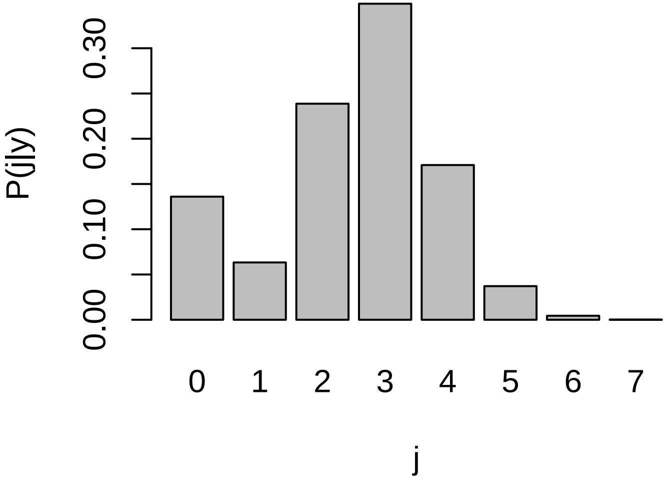


Figure 15: Posterior on the number of clusters

Bayesian cluster model

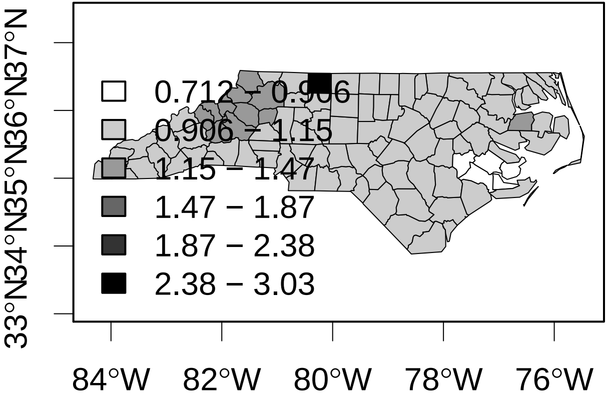


Figure 16: Posterior relative risk estimates