2023 554 R Notes for Assessment of Clustering for Count Data

Jon Wakefield
Departments of Biostatistics and Statistics
University of Washington

2023-03-01

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 are 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 for 1974.

The expected numbers are based on internal standardization with a single stratum. So the single reference probability is the incidence of SIDS in 1974.

```
library(maptools)
library(spdep)
nc.sids <- readShapeSpatial(system.file("shapes/sids.shp",
    package = "maptools")[1], IDvar = "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)
rm(nc.sids)  # We load another version of this later, so tidy up here
```

SMR Plot

The map of the SMRs shows a number of counties with high relative risks (the risk relative to the state wide risk).

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

Overdispersion

Examine κ, the overdispersion statistic.

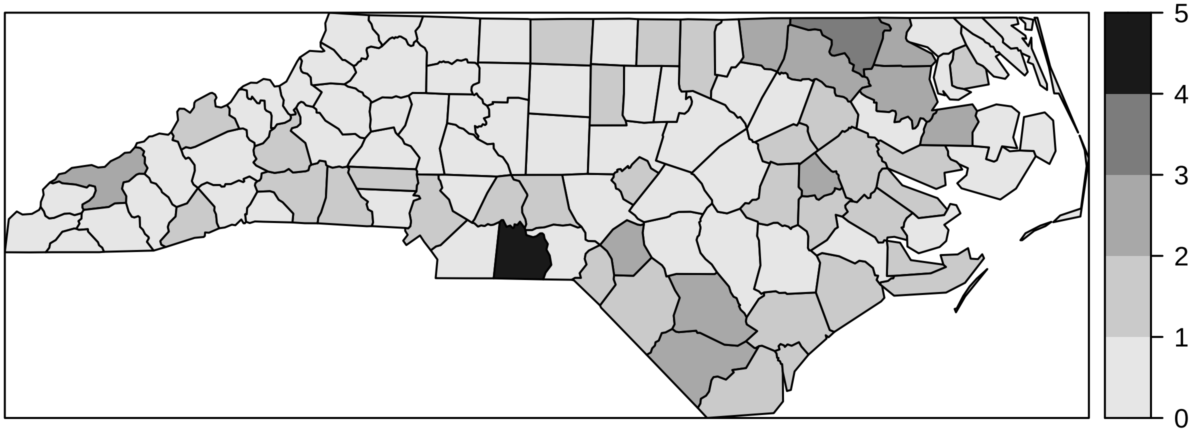


Figure 1: Map of SMRs for SIDS in 1974 in North Carolina

```
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)
cat("kappa = ", kappaest, "\n")
## kappa =  2.278508
```

Indicates extensive excess variation.

Disease Mapping

We first fit a non-spatial random effects model:

$$\begin{aligned} Y_i|\beta_0, \epsilon_i &\sim_{iid} \text{Poisson}(E_i e^{\beta_0+\epsilon_i}), \\ \epsilon_i|\sigma_\epsilon^2 &\sim_{iid} N(0, \sigma_\epsilon^2) \end{aligned}$$

with the default priors on β_0 and σ_ϵ^2.

```
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))
```

The `control.predictor` argument indicates we want fitted values.

Examine the first few “fitted values”, summaries of the posterior distribution of $\exp(\beta_0 + \epsilon_i), \qquad i = 1, \ldots, n$.

```
head(m0$summary.fitted.values)
##                           mean        sd 0.025quant  0.5quant 0.975quant
## fitted.Predictor.001 1.2500172 0.2980954  0.7697994 1.2147869   1.932168
## fitted.Predictor.002 0.7576456 0.2716406  0.3435563 0.7192293   1.396263
## fitted.Predictor.003 0.9156031 0.3526072  0.3999739 0.8592110   1.762773
## fitted.Predictor.004 2.7161635 0.7959618  1.4915354 2.6048776   4.578056
## fitted.Predictor.005 0.8910521 0.3181129  0.4139880 0.8434085   1.646841
## fitted.Predictor.006 0.8555516 0.3191849  0.3805787 0.8068988   1.615943
##                           mode
## fitted.Predictor.001 1.1483609
## fitted.Predictor.002 0.6507136
## fitted.Predictor.003 0.7648109
## fitted.Predictor.004 2.3919411
## fitted.Predictor.005 0.7610182
## fitted.Predictor.006 0.7234537
m0$summary.fixed
##                    mean         sd 0.025quant    0.5quant 0.975quant
## (Intercept) -0.02946208 0.06302482 -0.1565397 -0.02833104 0.09123525
##                    mode          kld
## (Intercept) -0.02610935 2.725421e-08
m0$summary.hyperpar
##                     mean       sd 0.025quant 0.5quant 0.975quant    mode
## Precision for ID 7.268522 2.567516   3.802461 6.775314   13.56196 6.00787
```

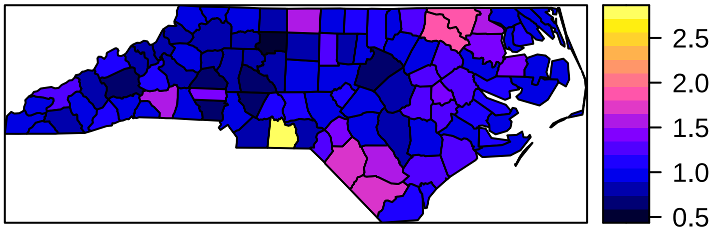
Create two interesting inferential summaries:

- the posterior mean of the relative risk
- a binary indicator of whether the posterior median is greater than 1.5 (which we assume is an epidemiologically significant value). This value can be changed, based on the context.

```
nc.sids2$RRpmean0 <- m0$summary.fitted.values[,
    1]
nc.sids2$RRind0 <- m0$summary.fitted.values[,
    4] > 1.5
```

Disease Mapping

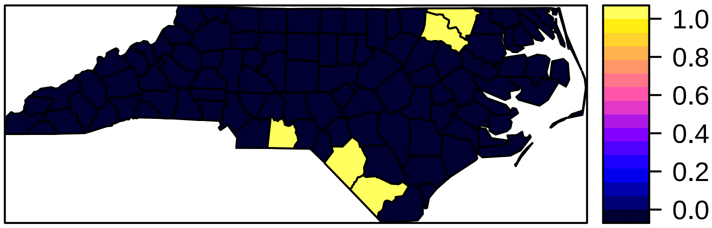
```
# Display posterior means of
# relative risks
spplot(nc.sids2, "RRpmean0")
```



A number of counties have high mean values.

Disease Mapping

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



Six counties have posterior median relative risk greater than 1.

Disease Mapping with IID and ICAR random effects

We now fit a model with non-spatial and ICAR spatial random effects, i.e. the BYM2 model.

```
m1 <- inla(SID74 ~ 1 + f(ID, model = "bym2", graph = "NC.graph"),
    family = "poisson", E = EXP74, data = as.data.frame(nc.sids2),
    control.predictor = list(compute = TRUE))
# Define summary quantities of interest as with
# iid model
nc.sids2$RRpmean1 <- m1$summary.fitted.values[, 1]
nc.sids2$RRind1 <- m1$summary.fitted.values[, 4] >
    1.5
m1$summary.fixed
##                    mean         sd 0.025quant    0.5quant 0.975quant
## (Intercept) -0.05004052 0.05831879 -0.1661376 -0.04949412 0.06292908
##                    mode          kld
## (Intercept) -0.04845334 8.606241e-09
m1$summary.hyperpar
##                       mean        sd 0.025quant  0.5quant 0.975quant      mode
## Precision for ID 6.2687958 2.2035941   3.122117 5.8779375 11.6770387 5.1711296
## Phi for ID       0.6596086 0.2272449   0.174615 0.6985815  0.9730934 0.9193931
```

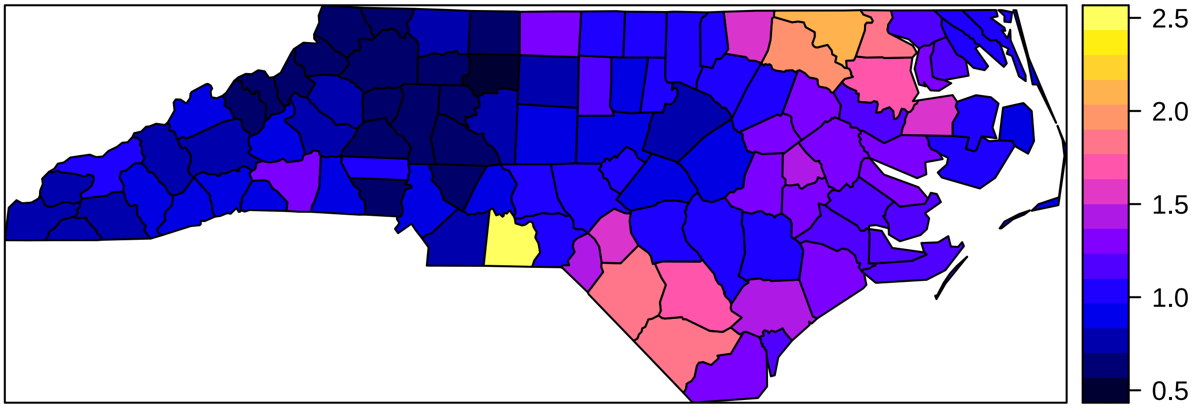
an aside, if we wanted to create a neighbour list based on regions with contiguous boundaries we can use the `poly2nb` function in the `spdep` library.

```
nc.sids <- readShapeSpatial(system.file("shapes/sids.shp", package = "maptools")[1],
    IDvar = "FIPSNO", proj4string = CRS("+proj=longlat +ellps=clrk66"))
# Create adjacency matrix
nc.nb <- poly2nb(nc.sids)
nb2INLA("inlanc.graph", nc.nb)  # Slighty different to NC.graph (islands?)
rm(nc.sids)
```

Disease Mapping with IID and ICAR random effects

Display posterior means of relative risks.

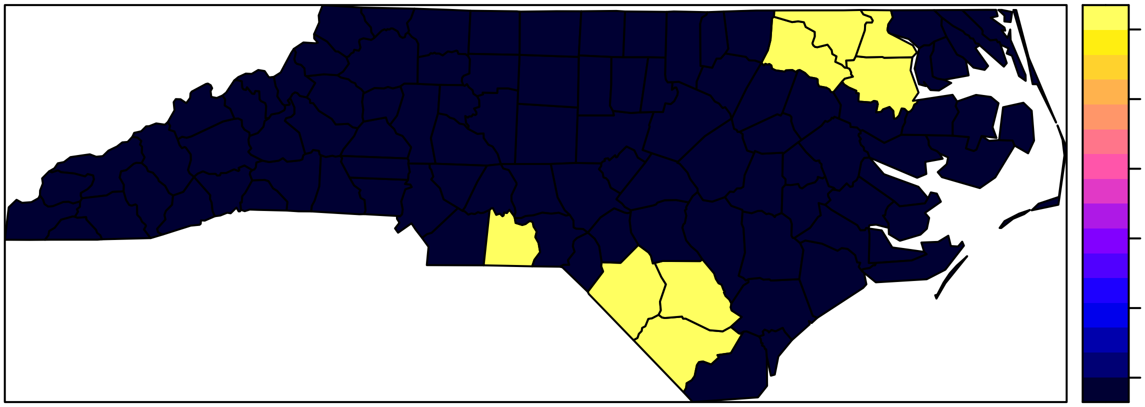
```
spplot(nc.sids2, "RRpmean1")
```



Disease Mapping with IID and ICAR random effects

Display areas with medians above 1.5, ie those areas with greater than 50% chance of exceedence of 1.5.

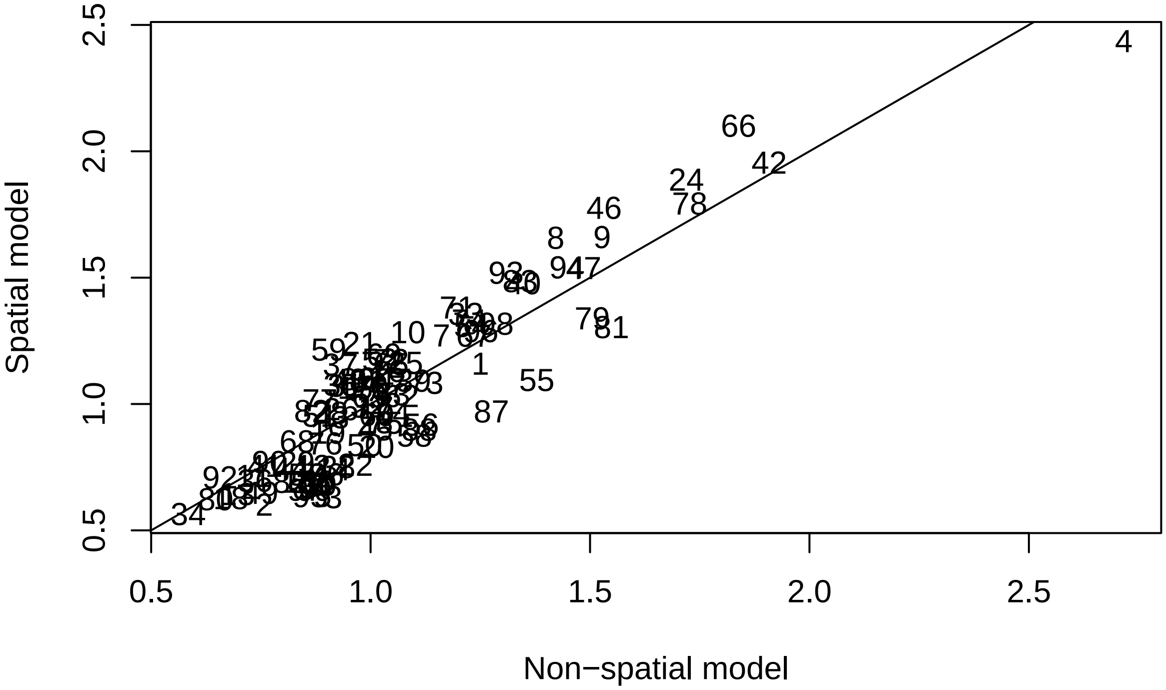
```
spplot(nc.sids2, "RRind1")
```



Both summaries show differences with the iid only model, with changes in the obvious direction. The spatial smoothing model gives a larger collection in the north-east, for example, and a single area in the west is not highlighted. As we will see later, the spatial model is supported by the data in this example.

Disease Mapping: Comparison of posterior mean RRs

```
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)
```



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. Also define the "B" style for later.

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

```
library(spdep)
# Note the nc.sids loaded from the data() command is in a different order to
# that obtained from the shapefile
data(nc.sids)
col.W <- nb2listw(ncCR85.nb, style = "W", zero.policy = TRUE)
col.B <- nb2listw(ncCR85.nb, style = "B", zero.policy = TRUE)
rm(nc.sids)
quasipmod <- glm(SID74 ~ 1, offset = log(EXP74), data = nc.sids2, family = quasipoisson())
sidsres <- residuals(quasipmod, type = "pearson")
```

```
moran.test(sidsres, col.W)
##
##  Moran 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
```

This analysis suggests significant clustering.

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

Hence, we should endeavor to remove the large-scale trends.

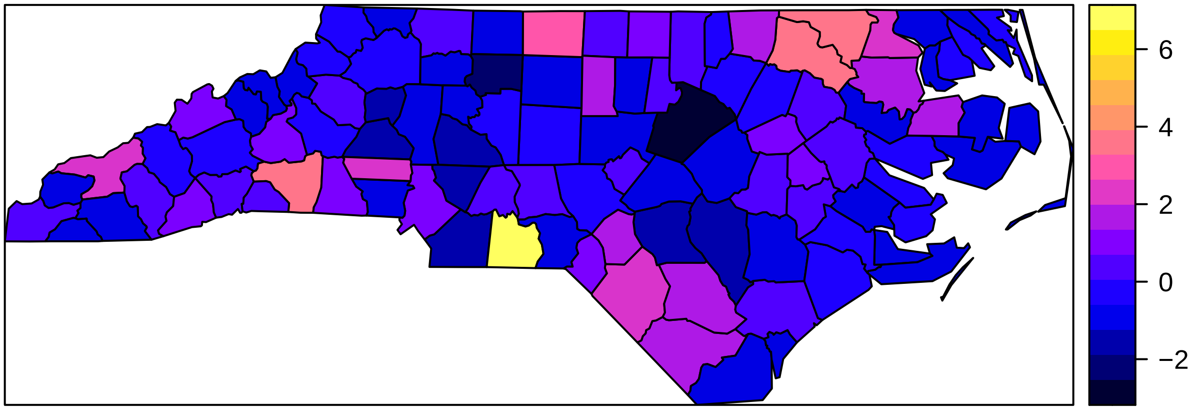
Below we fit a model with Eastings and Northings (of the County seat) as covariates – both show some at least some association.

```
# add east and north to the nc.sids2 data frame
data(nc.sids)
nc.sids2 <- merge(nc.sids2, nc.sids[, c("CNTY.ID", "east", "north")], by.x = "CNTY_ID",
    by.y = "CNTY.ID")
rm(nc.sids)

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 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2.039456)
##
##     Null deviance: 203.34  on 99  degrees of freedom
## Residual deviance: 171.80  on 97  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

We map the residuals to get a visual on the clustering.

```
sidsres2 <- residuals(quasipmod2, type = "pearson")
nc.sids2$res <- sidsres2
par(mar = c(0.1, 0.1, 0.1, 0.1))
spplot(nc.sids2, "res")
```



The significance of the Moran statistic is reduced, though still significant if judged by conventional levels.

```
moran.test(sidsres2, col.W)
##
##  Moran 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
```

Neighborhood options

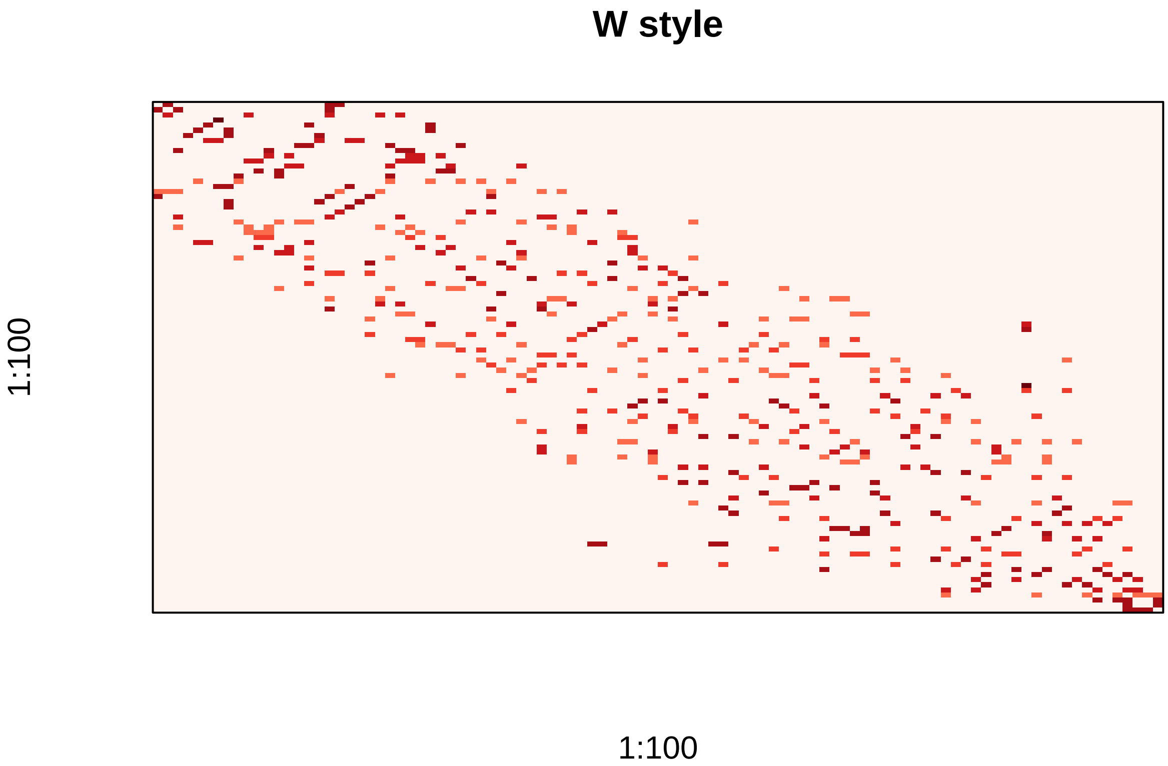
There are various coding schemes for the weights.

B has 0/1 corresponding to non-neighbor/neighbor – this means areas with many neighbors are more influential.

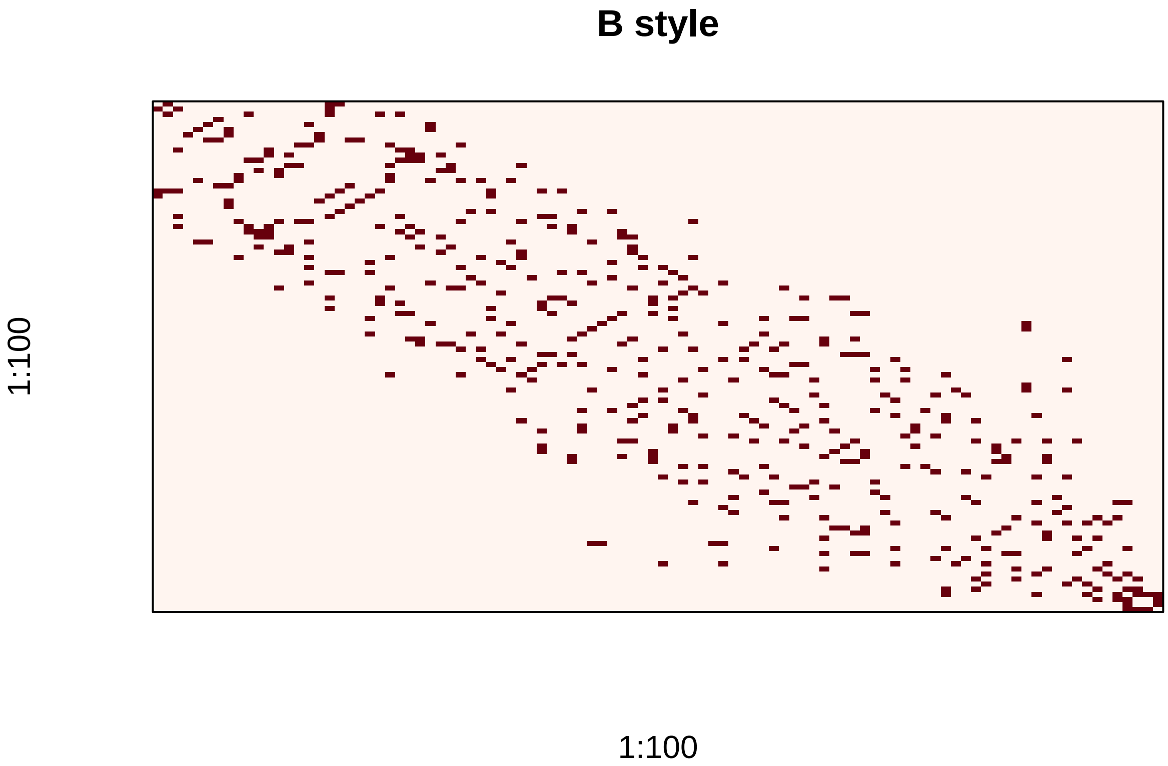
W has rows standardized by the number of neighbors so that the sum for each row (area) is unity.

Weights can be more complex, depending on inverse distance, for example. See Bivand et al. (2013, Section 9.2).

```
library(RColorBrewer)
pal <- brewer.pal(9, "Reds")
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()
```



```r
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()
```



Note the asymmetry in the "W" weights option.

Moran's I with a different neighborhood structure

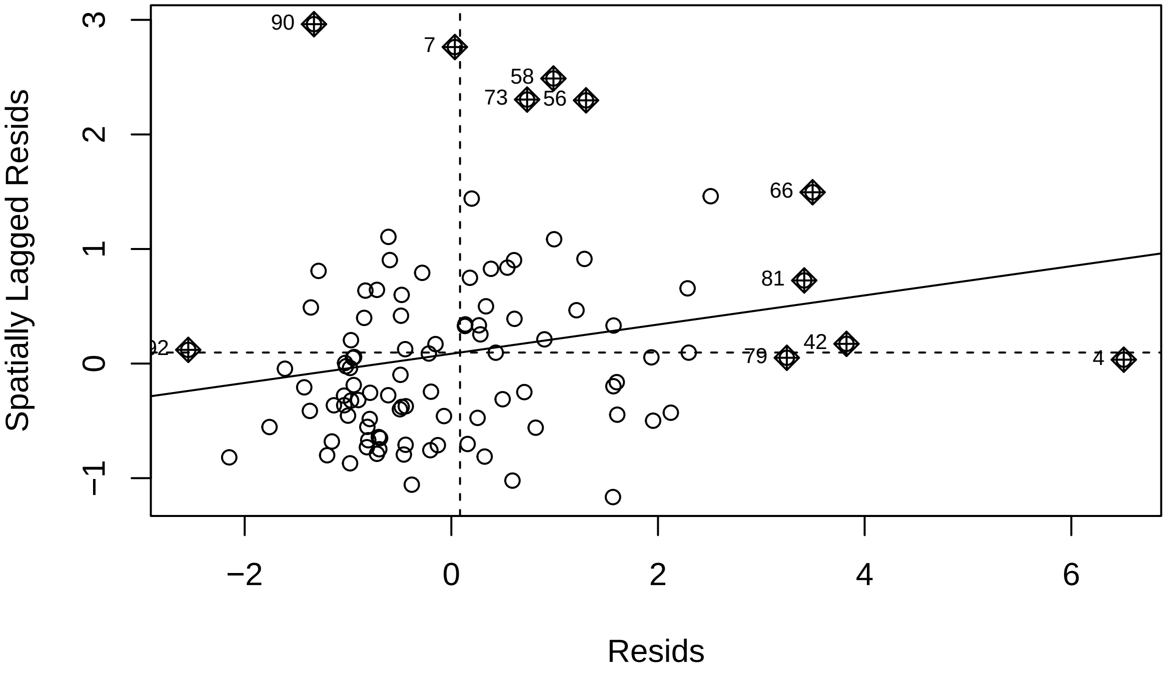
We now use Moran's statistic on the detrended residuals, but with the binary "B" weight option. This option has unstandardized weights.

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

```
moran.test(sidsres2, col.B)
##
##  Moran 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
```

Local influence

```
tmp <- matrix(c(rbind(1:length(sidsres2)), sidsres2), ncol = 2, nrow = length(sidsres2))
moran.plot(tmp[, 2], col.W, labels = tmp[, 1], xlab = "Resids", ylab = "Spatially Lagged Resids")
```



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 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 
```

North Carolina SIDS Data: Clustering Conclusions

The disease mapping model shows that almost all of the residual variation is spatial.

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

So all methods in agreement!